

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Teste e Certificação de Componentes para Ecosystema de *Ambient Assisted Living*

Bruno Miguel Carvalhido Lima



Mestrado Integrado em Engenharia Informática e Computação

Orientador: João Pascoal Faria (Doutor)

3 de Fevereiro de 2014

Teste e Certificação de Componentes para Ecossistema de *Ambient Assisted Living*

Bruno Miguel Carvalhido Lima

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: José Manuel de Magalhães Cruz

Arguente: José Maria Fernandes

Vogal: João Pascoal Faria

3 de Fevereiro de 2014

Este projeto é financiado pelo FEDER – Fundo Europeu de Desenvolvimento Regional através do programa COMPETE - Programa Operacional Factores de Competitividade no âmbito do projeto AAL4ALL - «FCOMP-01-0202-FEDER-013852» (mais detalhes em www.aal4all.org).



Resumo

Devido ao envelhecimento da população mundial é urgente que os sistemas tradicionais de saúde encontrem soluções que permitam aumentar o tempo de vida das pessoas no seu ambiente preferido, aumentando a sua autonomia, autoconfiança e mobilidade. É precisamente neste contexto que se insere o projeto AAL4AAL que, através do desenvolvimento de um ecossistema de produtos e serviços normalizados e interoperáveis de *Ambient Assisted Living* (AAL), pretende ser uma resposta para este problema.

Inserido no projeto AAL4ALL surge o presente projeto de dissertação, que teve como objetivo definir uma metodologia de teste e certificação para componentes de AAL. A metodologia proposta é assente num processo composto por três diferentes fases: análise de pré-requisitos, testes de conformidade e testes de interoperabilidade. Os pré-requisitos visam garantir que os componentes candidatos possuem evidências que são à partida necessárias para entrarem no ecossistema. Os testes de conformidade têm como objetivo garantir que o componente candidato comunica de acordo com aquilo que está especificado nas normas. Os testes de interoperabilidade visam comprovar que o componente candidato se comporta como é esperado quando integrado com outros componentes.

A metodologia proposta foi ainda alvo de uma validação através de um cenário piloto tão real quanto possível, composto por componentes do projeto AAL4ALL, para o qual foram definidos casos de testes e desenvolvida uma plataforma automática de teste, que permitiu não só automatizar alguns desses testes mas também otimizar o processo de teste e certificação.

No presente documento será apresentada, uma descrição detalhada do problema, um levantamento do estado da arte que engloba não só uma análise a *standards* na área da certificação, bem como a análise a iniciativas semelhantes na área da saúde. É também apresentada a metodologia de teste e certificação proposta e os resultados obtidos na sua aplicação prática no contexto do projeto AAL4ALL.

Abstract

Due to the aging of the world population it is urgent that traditional health systems find solutions to increase the people lifetime in their preferred environment by increasing their autonomy, self-confidence and mobility. It is precisely in this context that fits the AAL4AAL project that, through the development of an ecosystem of products and standardized and interoperable services of Ambient Assisted Living (AAL), aims to be an answer to this problem.

Inserted in the AAL4ALL project arises this dissertation project, which aimed to define a methodology for testing and certification components of AAL. The proposed methodology is based on a process composed of three different phases: analysis of prerequisites, conformance testing and interoperability testing. The prerequisites are intended to ensure that the candidates components have evidences that are necessary to enter the ecosystem. The conformance test, aim to ensure that the candidate component communicates according to what is specified in the rules. Interoperability tests aim to demonstrate that the candidate component behaves as expected when integrated with other components.

The proposed methodology was validated in a pilot scenario, composed of components of AAL4ALL project, for which test cases were defined. Was also set up an automatic testing platform, which allowed not only automate some of these tests but also optimize the testing and certification process.

In this document, it is presented a detailed description of the problem, a survey of the state of the art which includes not only an analysis of the standards in the area of accreditation as well as the analysis to similar initiatives in healthcare. It is also presented the proposed testing and certification methodology and the results obtained in practical application in the context of AAL4ALL project.

Agradecimentos

Esta dissertação não teria sido possível sem o apoio de várias pessoas.

Em primeiro lugar gostaria de agradecer ao meu orientador o Prof. Dr. João Pascoal Faria, que me deu todo o acompanhamento que eu precisei ao longo do último ano e que em alguns momentos foi mais que um mero orientador.

Agradeço também, aos meus colegas do Laboratório de Engenharia de Software, por terem estado sempre dispostos a ajudar em tudo o que eu precisei.

Gostaria ainda de agradecer aos meus professores da Escola Profissional de Vila do Conde, que sempre acreditaram em mim e nas minhas capacidades, me incentivando a prosseguir os estudos de forma a completar uma licenciatura. Pois bem, isto é mais que isso, esta é a minha tese de mestrado! Obrigado Prof. Manuela Paulo, Prof. Teresa Amaro e Prof. José Paulo Santos!

O meu muito obrigado a todos os meus professores, ao longo destes últimos cinco anos de curso que me ensinaram muito!

Tenho também de agradecer à minha família, principalmente aos meus pais, por me terem dado esta oportunidade, pois eu sei perfeitamente os sacrifícios que muitas vezes fizeram para que eu pudesse obter um grau académico. Apesar de eu as vezes não o demonstrar, vocês sabem... são os meus heróis!

Por último, e não menos importante, quero agradecer a minha namorada “Ni”, que esteve sempre do meu lado ao longo destes anos e que apesar de não gostar de informática teve muitas vezes de me ouvir falar de bits e bytes... A tua ajuda e compreensão ao longo destes anos foram fundamentais! “Ni, obrigado por tudo!”

Bruno Miguel Carvalhido Lima

“Testing is an infinite process of comparing the invisible to the ambiguous in order to avoid the unthinkable happening to the anonymous.”

James Bach

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Objetivos e Resultados Esperados	3
1.3	Metodologia	4
1.4	Estrutura do Documento	5
2	Revisão Bibliográfica	7
2.1	Normas para Avaliação de Conformidade e Qualidade	7
2.1.1	Família de Normas ISO/IEC 17000 para Avaliação da Conformidade . .	7
2.1.2	ISO/IEC 25000:2005 Engenharia de Software - Qualidade do Produto . .	12
2.2	Normas de Interoperabilidade para Dispositivos de Saúde	12
2.3	Procedimentos de Teste de Conformidade e Interoperabilidade	14
2.4	Iniciativas de Certificação na Área da Saúde e AAL	15
2.4.1	<i>Continua Health Alliance</i>	15
2.4.2	EHR-Q	22
2.4.3	HITCH	26
3	Proposta de Metodologia de Teste e Certificação	31
3.1	Processo de Teste e Certificação	31
3.2	Metamodelos e Templates	35
3.3	Estratégia de Teste	39
3.3.1	Fases do Processo de Teste	39
3.3.2	Abordagem Incremental de Configurações de Teste	41
3.4	Certificados	43
4	Aplicação e Validação da Metodologia	47
4.1	Cenário de teste	47
4.2	Análise da Documentação dos Produtos e Serviços	48
4.3	Redocumentação	49
4.4	Especificação dos Testes	51
4.5	Implementação dos <i>Scripts</i> de Testes	56
4.6	Resultados da Execução	60
4.7	Resultados do Processo de Certificação	63
5	Conclusões e Trabalho Futuro	65
5.1	Satisfação dos Objetivos	65
5.2	Trabalho Futuro	66
	Referências	67

CONTEÚDO

A	Artigos Produzidos	69
A.1	<i>A Testing and Certification Methodology for an Ambient-Assisted Living Ecosystem</i>	70
B	Plataforma de automatização de testes sobre o componente AALNode	75
C	Descrição dos componentes que integram o cenário de teste	85
C.1	<i>Template</i> de descrição de componentes do projeto AAL4ALL	86
C.2	Descrição do componente “AALNode” utilizando o <i>Template</i>	107

Lista de Figuras

1.1	Entidades que formam o consórcio AAL4ALL	2
1.2	Estrutura do Projeto AAL4ALL	2
1.3	Arquitetura do ecossistema definida na proposta do projeto AAL4ALL	4
2.1	ISO/IEC 17000 Normas para Avaliação da Conformidade	8
2.2	ISO/IEC 17000 abordagem funcional para avaliação da conformidade	10
2.3	Organização da série SQuaRE da ISO/IEC 25000	12
2.4	Contexto geral da norma ISO/IEEE 11073	13
2.5	Configuração geral de um teste de conformidade	14
2.6	Configuração geral de um teste de interoperabilidade	15
2.7	Dispositivos <i>Continua</i>	18
2.8	Arquitetura de Referência Ponta-a-ponta da <i>Continua</i>	18
2.9	Interfaces e <i>Standards</i> da <i>Continua</i>	19
2.10	Etapas do Processo de Certificação da <i>Continua</i>	19
2.11	Workflow da Certificação da <i>Continua</i>	21
2.12	Países Benefeciários do Projeto EHR-Q	22
2.13	Roteiro EHR-Q para a Certificação a Nível Nacional	23
2.14	Roteiro EHR-Q para a Certificação a Nível Europeu	24
2.15	Conceito do projeto HITCH	26
2.16	Processo de certificação/rotulagem do projeto HITCH	27
2.17	Relações entre as entidades envolvidas no processo de certificação	28
2.18	LabelCert - Cenários típicos	29
3.1	Responsabilidades da Associação AAL4ALL	32
3.2	Responsabilidades da Entidade Certificadora AAL4ALL	33
3.3	Processo de Certificação AAL4ALL	34
3.4	Diagrama de classes UML do metamodelo para a especificação de produtos e serviços AAL normalizados e os casos de teste associados	36
3.5	Diagrama de sequência UML para descrição das mensagens enviadas/recebidas pelos componentes de um determinado cenário de integração	37
3.6	Cenário exemplo	41
3.7	Interfaces do AALNode	42
3.8	Interfaces da Aplicação A	42
3.9	Exemplo de um certificado AAL4ALL	45
4.1	Cenário de teste	47
4.2	Representação da interface do script de execução de teste do “AALNode”	59
4.3	Representação da recepção de informação por parte do “AALNode”	59

LISTA DE FIGURAS

4.4	Representação da receção de informação por parte da aplicação “Caretaker Application”	60
4.5	Relatório de não conformidade emitido para o “AALNode”	64

Lista de Tabelas

2.1	ISO/IEC 17000 - Técnicas de avaliação da conformidade	11
2.2	Barreiras para tornar realidade o ecossistema de cuidados pessoais de saúde . . .	16
2.3	Expectativas para a certificação de produtos de TI para a saúde	17
2.4	Descrição das Etapas do Processo de Certificação da <i>Continua</i>	20
2.5	Recomendações do Projeto EHR-Q	25
3.1	Exemplo da descrição de uma categoria de componente	38
3.2	Descrição de um teste de conformidade segundo a norma IEEE Std 829 TM -2008 .	39
4.1	Requisitos identificados para o “AALNode”	49
4.2	Descrição mais formal do componente “AALNode”	50
4.3	Especificação dos testes de conformidade para o “AALNode”	52
4.4	Requisitos identificados para a aplicação “Caretaker Application”	53
4.5	Especificação dos testes de conformidade para a aplicação “Caretaker Application”	53
4.6	Requisitos identificados para o conjunto Gateway + Driver + Sensor	54
4.7	Especificação dos testes de interoperabilidade para o conjunto Gateway + Driver + Sensor	54
4.8	Requisitos identificados para o cenário de teste	54
4.9	Especificação dos testes de interoperabilidade para o cenário de teste	55
4.10	Resultado da execução dos testes sobre o “AALNode”	61
4.11	Resultado da execução dos testes sobre a aplicação “Caretaker Application” . . .	62
4.12	Resultado da execução dos testes sobre o conjunto Gateway + Driver + Sensor . .	62
4.13	Resultado da execução dos testes sobre o cenário	63

LISTA DE TABELAS

Lista de Listagens

4.1	Class Sender	56
4.2	Class Subscriber	57
4.3	Conjunto de funções auxiliares	58
4.4	Script do teste Nod.T7	60
B.1	Script do Menu	75
B.2	Script do teste Nod.T1	78
B.3	Script do teste Nod.T2	79
B.4	Script do teste Nod.T3	80
B.5	Script do teste Nod.T4	81
B.6	Script do teste Nod.T5	82
B.7	Script do teste Nod.T6	83

LISTA DE LISTAGENS

Abreviaturas e Símbolos

AAL	<i>Ambient Assisted Living</i>
EHR	<i>Electronic Health Record</i>
HCP	<i>Health Cluster Portugal</i>
I&D	Investigação e Desenvolvimento
IHE	<i>Integrating the Healthcare Enterprise</i>
OEM	<i>Original Equipment Manufacturer</i>
PPS	Produtos, Processos e Serviços
TIC	Tecnologias de Informação e Comunicação

Capítulo 1

Introdução

1.1 Contexto e Motivação

Atualmente as sociedades desenvolvidas enfrentam mudanças demográficas severas, pois a população mundial está a envelhecer. Esta situação deve-se à baixa taxa de natalidade, causada por opções de vida ou razões económicas e ao aumento da esperança média de vida, causada pela melhoria dos cuidados de saúde, que sofreram um grande avanço científico e tecnológico ao longo dos últimos anos. Este envelhecimento da população mundial tem sido acompanhado por estudos, nomeadamente pelo *United Nations Population Fund* que estima que só entre 2005 e 2050 o número de pessoas com mais de 60 anos irá triplicar [Fun09]. Embora este aumento do envelhecimento da população mundial não aparente ser um problema, ele acarreta consigo outras consequências que se revelam grandes desafios para os países desenvolvidos. Entre eles, está o aumento das pessoas com limitações físicas bem como a diminuição do número de jovens que neste momento apoiam os idosos nas suas tarefas do dia a dia. Por outro lado colocam-se também desafios importantes aos sistemas tradicionais de saúde, nomeadamente a necessidade urgente de encontrar soluções que permitam aumentar o tempo de vida das pessoas no seu ambiente preferido, aumentando a sua autonomia, autoconfiança e mobilidade.

Uma das possíveis soluções para estes problemas são as tecnologias de *Ambient Assisted Living* (AAL). No entanto, e apesar destas tecnologias já estarem presentes e serem utilizadas para diferentes objetivos, estas “primeiras ofertas” para os utilizadores secundários e primários são monolíticas e incompatíveis, tornando-se soluções dispendiosas e potencialmente insustentáveis. É neste contexto que surge o projecto AAL4ALL [AAL11], que apresenta uma possível resposta a estes problemas, pretendendo juntar os principais *stakeholders* (instituições públicas, indústria, organizações de utilizadores e instituições de I&D) para discutirem a definição de um conjunto básico de serviços de AAL de interesse geral. As entidades que formam o consórcio AAL4ALL estão representadas na Figura 1.1.

Introdução



Figura 1.1: Entidades que formam o consórcio AAL4ALL

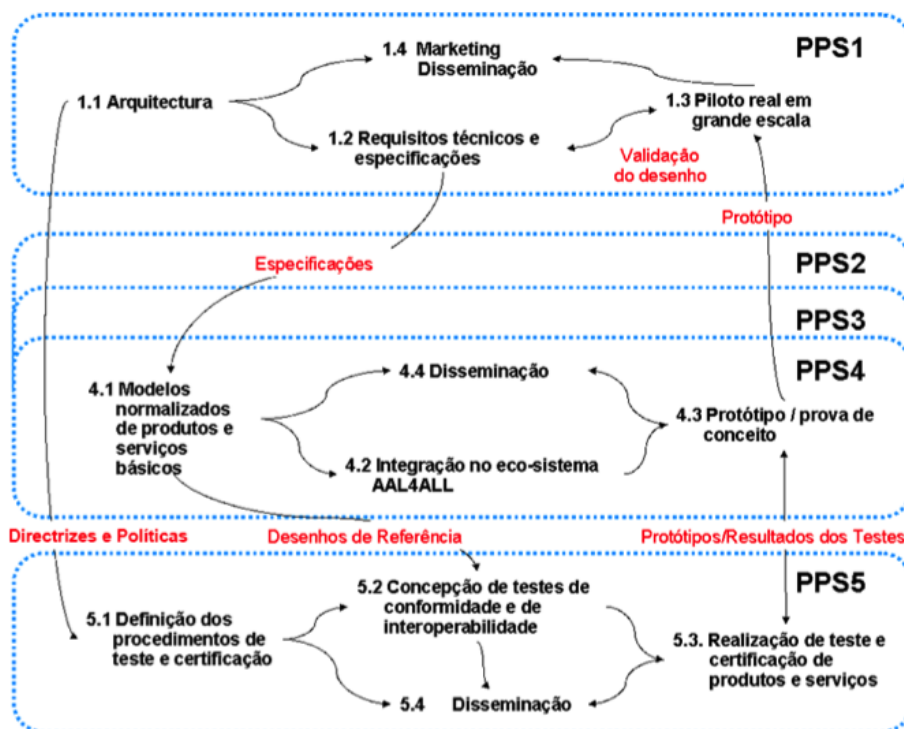


Figura 1.2: Estrutura do Projeto AAL4ALL

A análise das normas bem como de outras atividades internacionais já existentes são os princípios chave do projeto para evitar reinventar a roda, focando desta forma todos os esforços na implementação de soluções para as lacunas identificadas. O objetivo do projeto AAL4ALL é o desenvolvimento de um ecossistema de produtos e serviços de AAL com um modelo de negócio

associado e validado através de um piloto de grande escala. A definição de modelos de referência para diferentes ambientes servirá para diminuir o risco de investimento e o *time-to-market* de produtos e serviços desenvolvidos no futuro. A vantagem competitiva de *First mover* aliada à normalização destes produtos e serviços irá permitir a massificação de soluções AAL e a sua exportação para outros países, em particular, os do espaço europeu. O projeto é composto por cinco conjuntos de Produtos, Processos e Serviços (PPS), como representado na Figura 1.2, que de forma integrada deverão dar uma resposta aos desafios enunciados anteriormente. Os PPS foram definidos numa lógica de mercado com base nas especificidades dos produtos e serviços a serem desenvolvidos para os utilizadores.

Neste sentido os PPS2, PPS3 e PPS4 foram criados tendo como objetivo a segmentação dos produtos em 3 áreas diferentes: utentes de serviços de saúde; serviços de TIC e logística; entidades prestadoras de cuidados primários de saúde. O PPS1 funcionará como agregador de todos estes *stakeholders* e irá definir e validar a arquitetura global do ecossistema AAL4ALL. A Figura 1.3 apresenta um ponto de partida para a arquitetura do ecossistema definida na proposta do projeto AAL4ALL. Esta arquitetura pode, de uma forma simples, ser explicada como envolvendo duas principais entidades, os recetores de cuidados de saúde que podem ser monitorizados por sensor/atuadores, dentro ou fora das suas habitações. E os prestadores de cuidados de saúde, formais ou informais que recebem informações dos seus utentes através de aplicações que analisam dados. Toda esta infraestrutura é assente num *Bus* de comunicação que agrega e reencaminha toda a informação.

No PPS5 será elaborado o processo de certificação e o conjunto de testes a serem executados que devem garantir a interoperabilidade entre os diversos produtos e serviços e é neste contexto que surge a presente dissertação realizada no Instituto de Engenharia de Sistemas e Computadores do Porto.

1.2 Objetivos e Resultados Esperados

Nesta dissertação o objetivo principal é a criação de uma metodologia de teste e certificação para um ecossistema de AAL, aplicado neste caso ao projeto AAL4ALL. Desta forma pretende-se garantir a interoperabilidade e partilha de recursos e informações dentro do ecossistema, sendo que isso deverá ser conseguido através da normalização. Uma vez que este ecossistema pretende ser aberto, será também importante garantir que novos componentes ou serviços possam ser integrados de forma simples neste ecossistema. Estes objetivos apresentam vários desafios a serem superados, pois o ecossistema é composto por diferentes componentes, que possuem características específicas, desde sensores de monitorização até aplicações móveis, sendo não só importante garantir a comunicação entre dispositivos dentro de casa do receptor de cuidados de saúde até à garantia de comunicação ponta-a-ponta, desde o receptor de cuidados de saúde até ao fornecedor de cuidados de saúde, tendo em conta todas as necessidades que isso implica, não só ao nível da segurança dos dados, mas também aos tempos de resposta.

Introdução

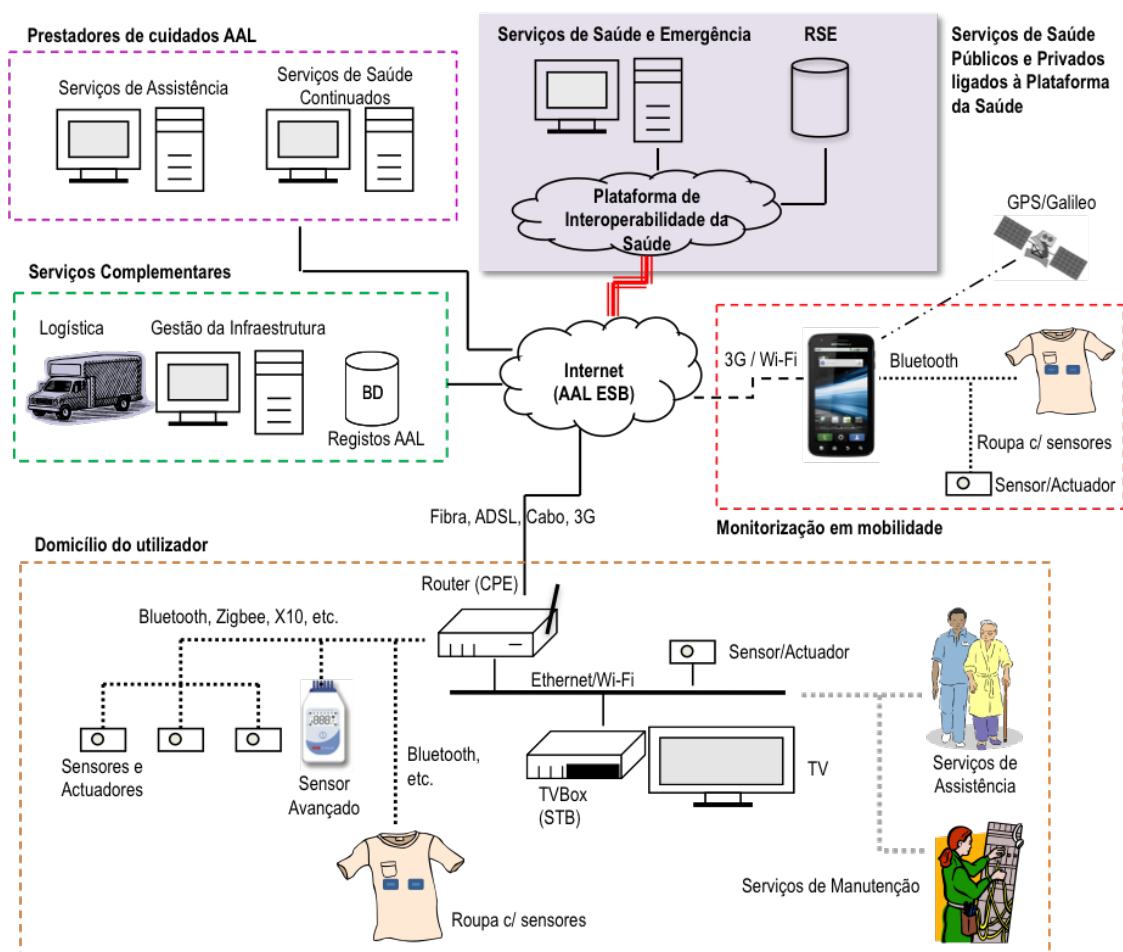


Figura 1.3: Arquitetura do ecossistema definida na proposta do projeto AAL4ALL

É ainda esperado o desenvolvimento de infra-estruturas de automação de testes, de suporte à operacionalização da metodologia definida.

A metodologia de teste e certificação proposta deve ser alvo de aplicação prática num ou mais cenários piloto, com vista à demonstração, validação e refinamento da metodologia, em interação com outros parceiros do projeto.

1.3 Metodologia

Por forma a atingir os objetivos propostos foram cumpridas as seguintes etapas. A primeira etapa passou pela análise do projeto AAL4ALL, tendo como objetivo compreender de forma clara o contexto e os requisitos deste projeto, bem como conhecer os parceiros envolvidos nas diferentes componentes. Uma vez que este é um projeto com diferentes intervenientes, é de extrema importância que as pessoas se conheçam, partilhem informação e trabalhem alinhados num objetivo comum. A segunda etapa passou por fazer um levantamento do estado da arte, não só de iniciativas existentes na área da saúde com objetivos ou características similares, mas também de

standards existentes que possam ser úteis na definição da metodologia de teste e certificação a desenvolver. Na terceira etapa foi definido o cenário concreto alvo de teste no âmbito deste projeto de dissertação, uma vez que só assim era possível avaliar concretamente o real resultado da metodologia proposta. A quarta etapa visou a definição da metodologia de teste e certificação sendo seguida de uma quinta etapa que teve como objetivo a criação e especificação de casos de teste e critérios de certificação, bem como a elaboração de uma plataforma de automação de testes. Na última etapa foram realizados os testes sobre o cenário, bem como interpretados os seus resultados, tendo ainda sido refinada a metodologia durante esse processo de teste.

1.4 Estrutura do Documento

Para além da introdução, este documento contém mais quatro capítulos. No capítulo ?? é descrito o estado da arte e são apresentadas iniciativas relacionadas na área da saúde, bem como enumerados algumas normas relevantes para o presente contexto. No capítulo 3 é apresentada a metodologia proposta e no capítulo 4 é demonstrada a aplicação dessa metodologia no contexto do projeto AAL4ALL. Por último, no capítulo 5 são apresentadas as conclusões sobre o trabalho realizado e descritas algumas orientações sobre o trabalho futuro.

Introdução

Capítulo 2

Revisão Bibliográfica

Neste capítulo é descrito o estado da arte e são apresentadas iniciativas relacionadas na área da saúde, bem como enumerados alguns *standards* que serão utilizados no decorrer do presente projeto.

2.1 Normas para Avaliação de Conformidade e Qualidade

2.1.1 Família de Normas ISO/IEC 17000 para Avaliação da Conformidade

Avaliação da conformidade é o nome dado aos processos que são usados para demonstrar que um produto (tangível), um serviço, um sistema de gestão ou um organismo obedece aos requisitos especificados. Essas exigências estão contidas em normas ISO/IEC e em guias tais como os processos que devem ser seguidos para que se possa comprovar a satisfação dos requisitos. A avaliação da conformidade pode abranger uma ou mais das seguintes atividades:

- Teste de um produto/serviço para determinar se ele cumpre os requisitos especificados;
- Verificação do processo de fabricação de um produto para assegurar que é fabricado de uma forma segura e de acordo com os regulamentos;
- Implementação de um sistema de gestão para garantir que os produtos/serviços são produzidos ou entregues por uma organização de uma forma consistente e que satisfaça as expectativas dos clientes.

Dentro da família de normas ISO/IEC 17000, encontram-se normas específicas para diferentes áreas (ver Figura 2.1), nomeadamente:

- **ISO/IEC 17000:2004 Avaliação da Conformidade - Vocabulário e princípios gerais [ISO04e]:** especifica disposições gerais e definições relativas à avaliação da conformidade, incluindo a

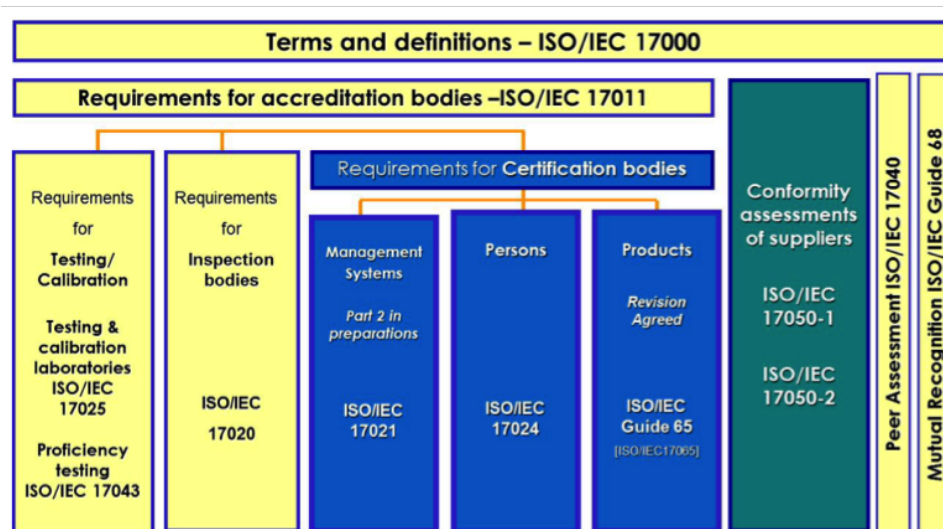


Figura 2.1: ISO/IEC 17000 Normas para Avaliação da Conformidade

acreditação de organismos de avaliação da conformidade. Também inclui um anexo informativo descrevendo uma abordagem funcional de avaliação da conformidade para dar um melhor entendimento das noções apresentadas.

- **ISO/IEC 17011:2004 Requisitos para organismos de acreditação [ISO04b]:** especifica os requisitos gerais para organismos de acreditação e avaliação e para acreditação dos organismos de avaliação da conformidade (OAC). Também define requisitos para o processo de avaliação por pares para acordos de reconhecimento mútuo entre organismos de acreditação.
- **ISO/IEC 17020:1998 Critérios gerais para o funcionamento de vários tipos de organismos de inspeção [ISO98]:** especifica os critérios gerais para a competência de organismos imparciais que realizam inspeções independentemente do sector envolvido. Também especifica critérios de independência.
- **ISO/IEC 17021:2006 Avaliação de conformidade - Requisitos para organismos de auditoria e certificação de sistemas de gestão [ISO06]:** contém princípios e requisitos para a competência, coerência e imparcialidade da auditoria e certificação de sistemas de gestão de todos os tipos (por exemplo, sistemas de gestão da qualidade ou sistemas de gestão ambiental) e para organismos que prestam essas atividades. Organismos de certificação que operam com esta norma internacional não precisam de oferecer todos os tipos de certificação do sistema de gestão.
- **ISO/IEC 17024:2003 Avaliação de conformidade - Requisitos gerais para organismos que realizam certificação de pessoas [ISO03]:** especifica requisitos para um organismo de certificação de pessoas no que diz respeito a requisitos específicos, incluindo o desenvolvimento e manutenção de um sistema de certificação de pessoal.

- **ISO/IEC 17025:2005 Requisitos gerais para a competência de laboratórios de ensaio e calibração [ISO05b]:** especifica os requisitos gerais para a competência de realização de testes e/ou calibrações, incluindo amostragem. É aplicável a todas as organizações que realizam ensaios e/ou calibrações. Estas incluem, por exemplo, os laboratórios de primeira, segunda e terceira parte, e laboratórios onde se realizam os testes e/ou calibração como formas de inspeção e certificação de produtos.
- **ISO/IEC 17030:2009 Avaliação de conformidade - Requisitos gerais para terceiras-partes em marcas de conformidade [ISO09]:** O principal objetivo desta norma é permitir uma abordagem uniforme para o uso de terceiros em marcas de conformidade, preencher lacunas importantes nas normas ISO, IEC, ISO/IEC e guias existentes, tratar de problemas potenciais decorrentes de diferentes usos de terceiras-partes em marcas de conformidade, fornecer uma base clara e racional para o seu uso, e estabelecer requisitos gerais. Esta norma concentra-se em terceiras-partes, mas também pode ser usada como orientação para outras aplicações de marcas de conformidade.
- **ISO/IEC 17050-1:2004 Avaliação da conformidade - Declaração de conformidade do fornecedor - Parte 1: Requisitos gerais [ISO04d]:** especifica os requisitos gerais para a declaração de conformidade do fornecedor em casos em que é desejável, ou necessário, que a conformidade de um objeto com os requisitos especificados seja atestado, independentemente do sector envolvido. Para os efeitos desta norma, o objeto de uma declaração de conformidade pode ser um produto, processo, sistema de gestão, pessoa ou organismo.
- **ISO/IEC Guia 28:2004 - Orientações gerais para um sistema específico de certificação de produtos [ISO04c]:** É aplicável a um sistema de certificação de produto de terceiros para determinar a conformidade de um produto com os requisitos especificados, através de testes iniciais a amostras do produto, avaliação e fiscalização do sistema de qualidade envolvido, e vigilância através de análises de amostras de produtos retirados da fábrica, do mercado livre, ou em ambos. Aborda ainda as condições de uso de uma marca de conformidade e as condições para a concessão de um certificado de conformidade.
- **ISO/IEC Guia 53:2005 - Orientação sobre o uso do sistema de gestão da qualidade de uma organização na certificação de produtos [ISO05a]:** Apresenta uma abordagem geral através da qual os organismos de certificação podem desenvolver e aplicar esquemas de certificação de produtos, utilizando os requisitos do sistema de gestão de qualidade de uma organização. As disposições dadas não são requisitos para a aprovação de um organismo de certificação de produtos e não substituem os requisitos da ISO/IEC Guia 65 [ISO96].
- **ISO/IEC Guia 65:1996 - Requisitos gerais para organismos que operam sistemas de certificação de produtos [ISO96]:** Pode ser usado em combinação com uma série de normas relacionadas com produtos e guias para demonstrar que um produto está em conformidade com os critérios especificados. Certificação de produto pode consistir em testes iniciais de um produto combinados com a avaliação do sistema de gestão da qualidade do

seu fornecedor. Isto pode ser seguido por ensaios de amostras a partir da fábrica e/ou do mercado. Outros esquemas de certificação de produtos compreendem testes iniciais e testes de vigilância, enquanto outros ainda contam com o teste de uma amostra do produto, também conhecido como ensaio de tipo.

- **ISO/IEC Guia 67:2004, Avaliação da Conformidade - Fundamentos de certificação de produto [ISO04a]:** Descreve algumas das atividades de certificação de produtos, identifica os elementos básicos e os tipos de certificação de produtos, e mostra algumas das formas de combinar esses elementos para projetar um sistema de certificação do produto.

Deste conjunto de normas é importante destacar as técnicas de avaliação da conformidade definidas pela norma ISO/IEC 17000 [ISO04e]. Estas podem ser realizadas por uma “primeira parte”, por uma “segunda parte” ou por terceiros, e estabelecem uma “abordagem funcional” que envolve o processo básico que é apresentado na Figura 2.2 e na Tabela 2.1.

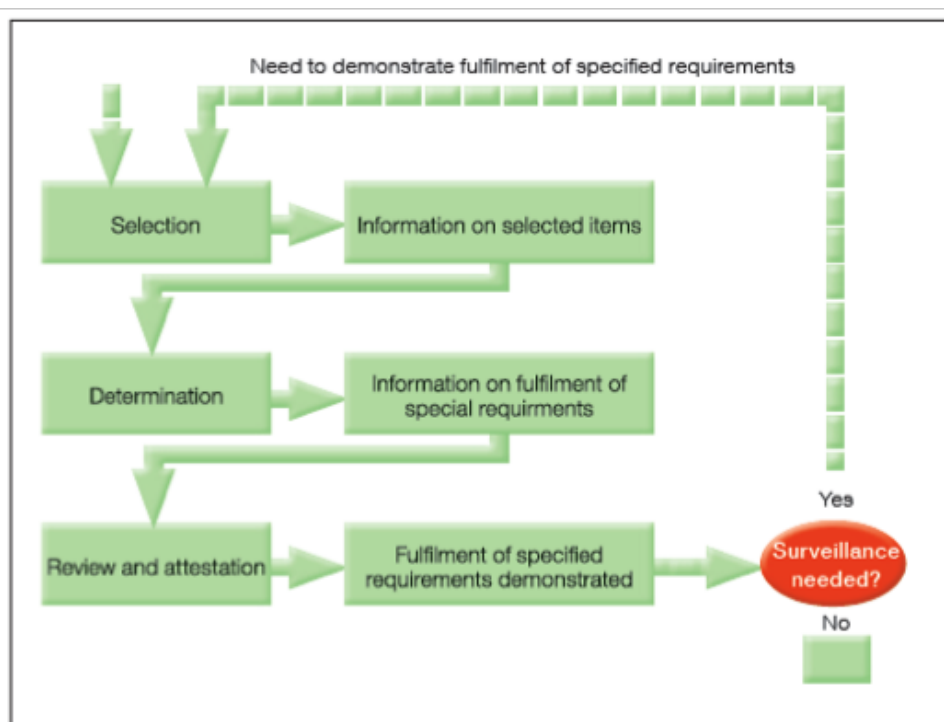


Figura 2.2: ISO/IEC 17000 abordagem funcional para avaliação da conformidade

A informação fornecida por esta família de normas apresenta um conjunto de recomendações úteis para a metodologia de teste e certificação a ser desenvolvida por este projeto, nomeadamente com a abordagem funcional e as técnicas de avaliação de conformidade apresentadas anteriormente.

Tabela 2.1: ISO/IEC 17000 - Técnicas de avaliação da conformidade

Técnicas ISO/IEC 17000	Descrição
Seleção	<ul style="list-style-type: none"> • Especificação das normas ou outros documentos que servem de base à avaliação de conformidade; • Seleção dos exemplares do objeto que está a ser avaliado; • Especificação de técnicas de amostragem estatística, se aplicável.
Determinação	<ul style="list-style-type: none"> • Teste para determinar características especificadas do objeto a ser avaliado; • Inspeção das características físicas do objeto a ser avaliado; • Auditoria dos sistemas e registos relativos ao objeto a ser avaliado; • Avaliação das qualidades do objeto a ser avaliado; • Análise de especificações e desenhos para o objeto a ser avaliado.
Revisão e Certificação	<ul style="list-style-type: none"> • Revisão das evidências recolhidas desde a fase de determinação quanto à conformidade do objeto com os requisitos especificados; • Retorno à fase de determinação para resolver as não-conformidades; • Elaboração e emissão de uma declaração de conformidade; • Colocar a marca de conformidade nos produtos certificados.
Vigilância	<ul style="list-style-type: none"> • Realização de atividades de determinação no ponto de produção ou na cadeia de abastecimento para o mercado; • Realização de atividades de determinação no mercado; • Realização de atividades de determinação no local de uso; • Analisar o resultado das atividades de determinação; • Retorno à fase de determinação para resolver as não-conformidades; • Elaboração e emissão de confirmação de que mantém a conformidade; • Iniciar correções e prevenções no caso de não-conformidades.

2.1.2 ISO/IEC 25000:2005 Engenharia de Software - Qualidade do Produto

A guia ISO/IEC 25000:2005, Engenharia de Software - Requisitos e Avaliação da Qualidade de Produtos de Software (do inglês SQuaRE) [ISO04a], fornece orientações para a utilização da nova série de normas internacionais chamadas Requisitos e Avaliação da Qualidade de Produtos de Software (Software Quality Requirements and Evaluation - SQuaRE). O objetivo deste guia é fornecer uma visão geral dos conteúdos, modelos de referência e definições comuns, bem como a relação entre os documentos, permitindo que utilizadores do guia adquiram um bom entendimento da série de normas internacionais, de acordo com a sua necessidade. Esta família de normas possui diferentes divisões, como representado na Figura 2.3.

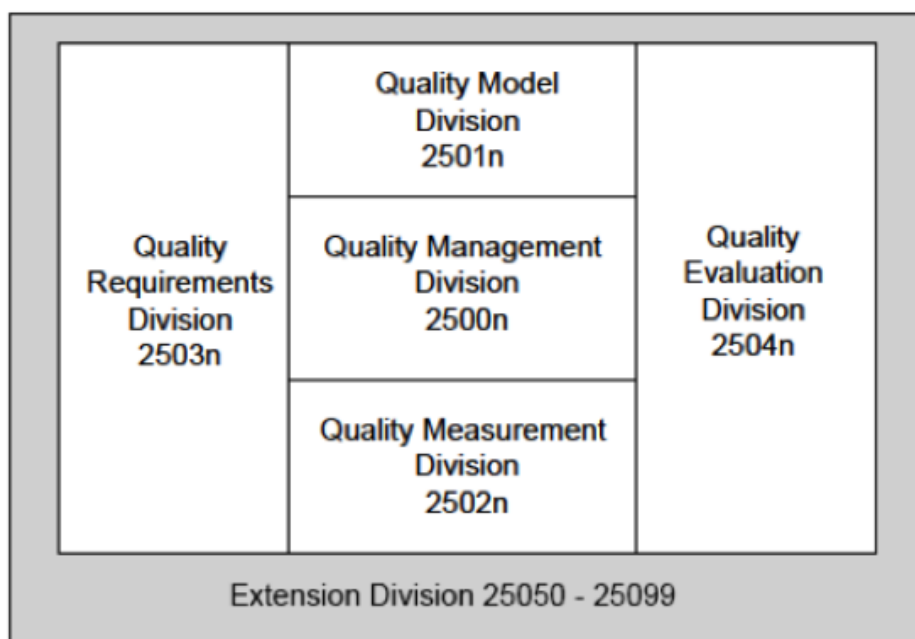


Figura 2.3: Organização da série SQuaRE da ISO/IEC 25000

De entre as várias divisões aquelas que pode ser úteis ao presente projeto são a ISO/IEC 2502n – Medição de Qualidade, pois contém um modelo de referência para medição da qualidade dos produtos de software e definição de métricas para medir a qualidade que podem ser utilizadas no processo de certificação a ser proposto e a ISO/IEC 2501n – Modelo de Qualidade, que entre outras informações apresenta um conjunto de regras para a classificação de requisitos.

2.2 Normas de Interoperabilidade para Dispositivos de Saúde

As normas da família ISO/IEEE 11073 permitem a comunicação entre dispositivos médicos e sistemas informáticos externos [ISO04f]. Esta norma e a norma correspondente IEEE 11073-104zz, abordam a necessidade de uma abordagem de comunicação simplificada e otimizada para dispositivos pessoais de saúde, os quais podem ou não ser dispositivos regulados. Estas normas

estão alinhadas com as normas existentes para equipamentos médicos, focadas para fornecer uma gestão fácil de dados a partir de qualquer dispositivo, quer ele seja um dispositivo pessoal, ou um dispositivo de uma instituição médica. A Figura 2.4 mostra as categorias e dispositivos típicos de apoio ao espaço de saúde pessoal. Agentes (por exemplo, monitores de pressão arterial, balanças, pedómetros) recolhem informações sobre uma pessoa (ou pessoas) e transferem as informações para o dispositivos gestores (por exemplo, telemóvel, aparelho de saúde, ou computador pessoal) para a recolha, visualização e possível transmissão posterior dessa informação. O dispositivo gestor também pode transmitir os dados para serviços de suporte remoto para análise posterior. Tais como serviços de gestão de doenças, gestão de dietas e *fitness*, ou aplicações para serviços de apoio ao envelhecimento de forma independente.

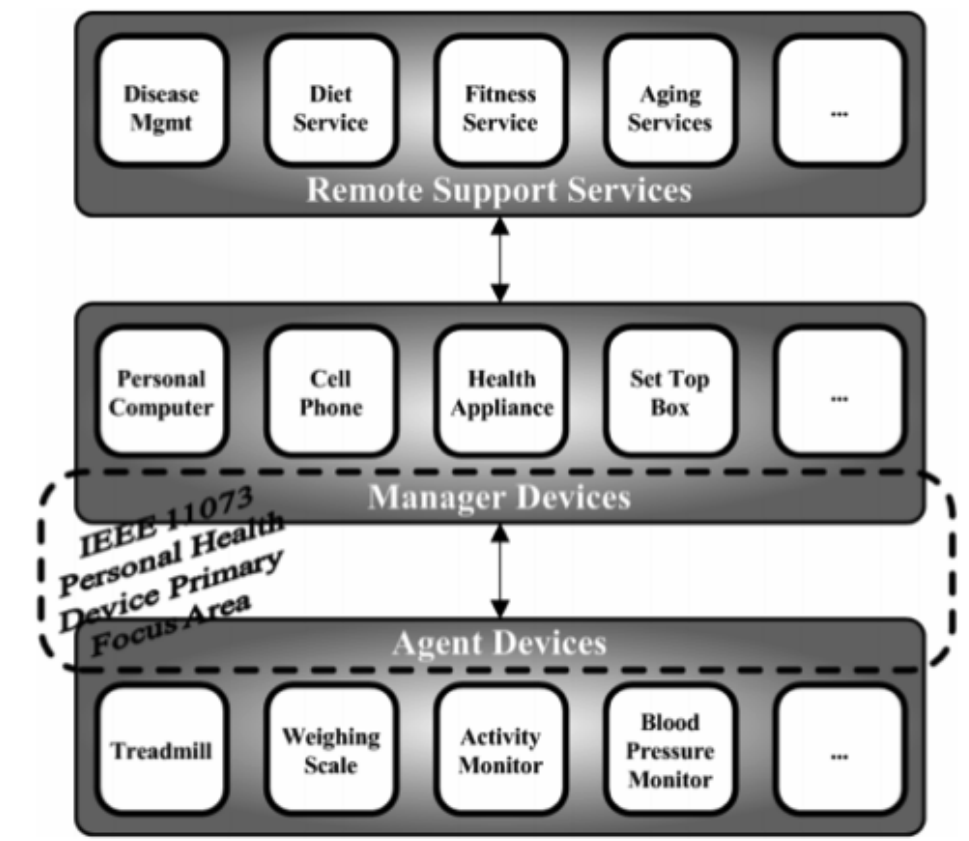


Figura 2.4: Contexto geral da norma ISO/IEEE 11073

Por ser uma norma já bastante utilizada na área dos dispositivos para a saúde, ela é importante para o presente projeto. O cumprimento desta norma será um dos pré-requisitos exigidos aos equipamentos pessoais de saúde que queiram receber a certificação AAL4ALL.

2.3 Procedimentos de Teste de Conformidade e Interoperabilidade

No contexto do teste e certificação de sistemas em relação a protocolos de comunicação padrão, existem dois tipos diferentes, mas complementares, de especificações de teste, que seguem o princípio: testar primeiro os componentes e depois testar o sistema [Ins13], são eles:

- Especificações de testes de conformidade;
- Especificações de testes de interoperabilidade

Quando o objetivo é comprovar a conformidade e a interoperabilidade, a solução é oferecer testes de conformidade em conjunto com testes de interoperabilidade [Ins07].

Os testes de conformidade são realizados para comprovar que um determinado produto ou serviço implementa um protocolo específico de acordo com os requisitos especificados para o protocolo em si [Ins07]. O teste de conformidade é caracterizado pelos seguintes fatores [Ins07]:

1. O sistema ou implementação a ser testado (SUT ou IUT) determina os limites para o teste.
2. O sistema de teste de conformidade executa os testes, controla o SUT e observa todas as comunicações do SUT. Os pontos onde o “testador” controla e observa o IUT são chamados de Pontos de Controle e Observação (PCO).
3. Os testes são realizados em interfaces abertas e normalizadas que são especificadas no nível de protocolo (que não são geralmente acessíveis para o utilizador normal).

A Figura 2.5 mostra um exemplo de um teste de conformidade.

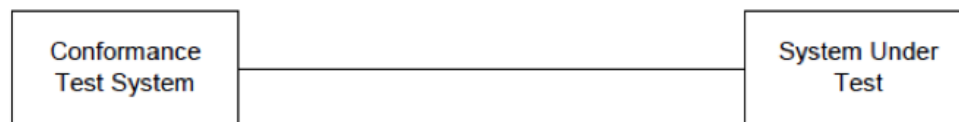


Figura 2.5: Configuração geral de um teste de conformidade

Os testes de interoperabilidade são realizados com o objetivo de provar que os produtos ou serviços de diferentes fabricantes são capazes de trabalhar juntos, usando um protocolo padrão [Ins07].

O teste de interoperabilidade é caracterizado pelos seguintes fatores [Ins07]:

1. O equipamento sob ensaio (EUT) e o equipamento qualificado (QE) é responsável pela definição dos limites para o teste.
2. Os fornecedores de equipamentos devem ser diferentes ou, pelo menos, o EUT e o QE devem ser provenientes de diferentes linhas de produtos.

3. Os testes de interoperabilidade são efetuados nas interfaces que permitem apenas o controlo pelo utilizador normal e observação.
4. Os testes de interoperabilidade não são baseados no nível de protocolo. A funcionalidade experimentada por um utilizador é a base deste tipo de teste, quer o utilizador seja um humano ou uma aplicação de software.
5. Os testes de interoperabilidade só são capazes de especificar o comportamento funcional, pois estes testes são realizados em interfaces funcionais e nos pontos finais. Com este tipo de teste não é possível claramente provocar ou provar um comportamento de erro do protocolo.

A figura a seguir (Figura 2.6) mostra um exemplo de teste de interoperabilidade.



Figura 2.6: Configuração geral de um teste de interoperabilidade

Ambos os tipos de testes apresentados são, pelas suas características, úteis para a metodologia de teste de certificação que se pretende desenvolver neste projeto.

2.4 Iniciativas de Certificação na Área da Saúde e AAL

2.4.1 *Continua Health Alliance*

A *Continua Health Alliance* [All13b] é uma organização sem fins lucrativos, fruto da união da indústria de cuidados de saúde e empresas de tecnologia de todo o mundo com o objetivo de colaborarem para melhorar a qualidade dos cuidados pessoais de saúde. Através dos esforços de uma organização industrial colaborativa, a *Continua* tem como objetivo possibilitar a criação de um ecossistema de saúde pessoal, onde muitos fornecedores diferentes podem combinar os seus produtos em novas propostas de valor, com benefícios significativos para a saúde das pessoas em todo o mundo. Esta organização identificou as seguintes barreiras (Tabela 2.2) para tornar real o ecossistema pessoal de saúde.

Tabela 2.2: Barreiras para tornar realidade o ecossistema de cuidados pessoais de saúde

Técnicas	Falta de interoperabilidade de dados; Falta de componentes da plataforma de interoperabilidade.
Regulamentares	Normas de segurança e de eficiência restringem o desenvolvimento de soluções de vários fornecedores.
Financeiras	Valor económico de novos modelos é difícil de demonstrar; É difícil chegar a um modelo de negócio.

A *Continua Health Alliance* aborda estas barreiras através das seguintes atividades [LRL⁺08]:

- **Guias de Desenho:** permitir sensores interoperáveis, plataformas de tele-saúde e serviços de saúde e bem-estar.
- **Certificação e Logótipos:** criar um logótipo reconhecível pelo consumidor que signifique a promessa de interoperabilidade com outros produtos certificados.
- **Regulamento de Saúde e Segurança:** trabalhar com as entidades reguladoras de forma a assegurar e gerir eficazmente soluções de diferentes fornecedores.
- **Reembolsos:** trabalhar com líderes nas indústrias de saúde para desenvolver novas maneiras de abordar os custos do fornecimento de sistemas pessoais de eHealth.

As guias de desenho da *Continua* [LRL⁺08] especificam requisitos para as interfaces de conectividade de dispositivos num Sistema de Saúde Pessoal. Elas especificam como utilizar as normas existentes para criar soluções interoperáveis de cuidados pessoais de saúde. É importante notar que as guias de desenho da *Continua* são apenas sobre as interfaces de conectividade. Elas referem-se ao transporte de rede, camada de mensagens e formato de dados a ser utilizado. As guias de desenho da *Continua* não tratam de serviços de gestão, técnicas de sensoriamento, técnicas de extração de parâmetros, etc. Estas guias foram escritas especificamente para fabricantes de dispositivos que pretendem passar pelo processo de certificação da *Continua* com seus dispositivos, para as empresas que integram dispositivos da *Continua* em sistemas e subsistemas, e para laboratórios de testes que certificam a conformidade com as especificações da *Continua*. Um produto certificado pela *Continua* garante algum nível de interoperabilidade que pode ser usado como um incentivo dos vendedores para comercializar com sucesso o seu produto (Tabela 2.3).

Tabela 2.3: Expectativas para a certificação de produtos de TI para a saúde

Prestadores de Cuidados	Assegurar de forma comprovada a utilidade clínica da aplicação de acordo com a sua finalidade. Querer receber o produto com as capacidades de interoperabilidade que lhe foram prometidos.
Pacientes	Têm preocupações de segurança (por exemplo, para dispositivos) e esperam alta qualidade de atendimento. Têm preocupações de privacidade e segurança dos dados de saúde.
Autoridades de Saúde	Esperam melhorar a qualidade e reduzir o custo da assistência ao paciente, promovendo a adoção destes produtos de forma proativa ou através de regulamentação. Precisam de produtos com utilidade clínica comprovada e garantia de benefícios esperados.
Vendedores	Precisam de um mercado harmonizado e estável, de fácil acesso e economicamente eficiente de modo a fornecer soluções acessíveis e de alta qualidade. Pretendem promover a adoção de tais produtos no processo de prestação de cuidados, por meio de incentivos relacionados com a certificação.
Organizações de normalização	Promover o desenvolvimento e implantação de normas

A arquitetura de referência da *Continua* [LRL⁺08, CCSS07] define três tipos de elementos:

- **Componentes:** modelos de entidades lógicas, tais como serviços.
- **Dispositivos:** modelos de entidades físicas. Cada dispositivo pode hospedar um ou mais componentes.
- **Interfaces:** definem a comunicação entre os componentes. Interfaces entre os componentes hospedados por diferentes dispositivos são denominadas de interfaces de rede, enquanto que as interfaces entre os componentes hospedados pelo mesmo dispositivo são chamados de interfaces de programação de aplicações (APIs).

Dependendo das interfaces implementadas, os dispositivos representados na Figura 2.7, são classificados como:

- **Application Hosting Device:** dispositivo, como um computador pessoal, telemóvel, ou *hub* de monitorização capaz de alojar aplicações.
- **Personal Area Network (PAN) Device:** um sensor ou um atuador.
- **Local-Area Network (LAN) Device:** Agrega e partilha (numa rede) a informação dos dispositivos PAN a ele ligados (isto é frequentemente referido como uma função de *proxy*). Um dispositivo de LAN também pode implementar a funcionalidade de sensores e atuadores diretamente.
- **Wide Area Network Device:** implementa um serviço de gestão baseada em rede, usando recursos de comunicação IP (como xDSL, DOCSIS, GPRS, EDGE).
- **Health Record Network Device:** implementa um registo de saúde.

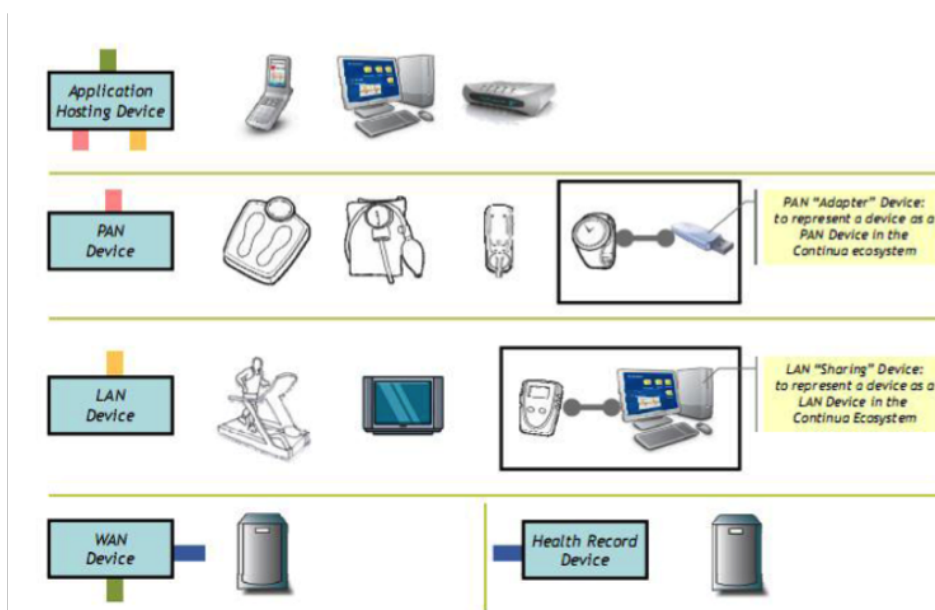


Figura 2.7: Dispositivos *Continua*

As interfaces de ligação oferecidas por um dispositivo determinam a que outros dispositivos pode ser ligado. Por outras palavras, existem algumas limitações na topologia baseada nas interfaces de ligação de um dispositivo. A topologia de referência da *Continua*, representada na Figura 2.8, mostra que um *PAN Device* pode ser ligado a um *Application Hosting Device* e este último pode ser ligado a um *WAN Device*.

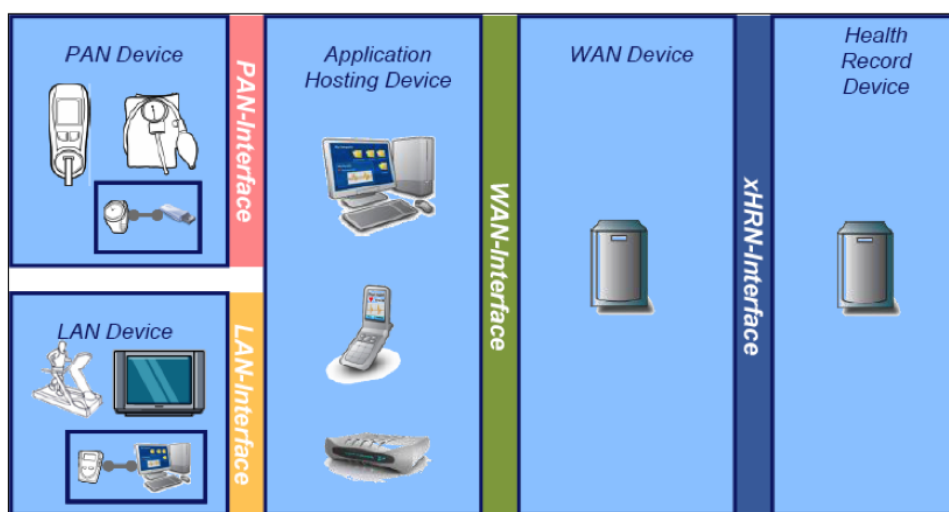


Figura 2.8: Arquitetura de Referência Ponta-a-ponta da *Continua*

As interfaces de rede estão no centro dos objectivos de interoperabilidade da *Continua* e são o ponto crucial das metas de teste e certificação para os dispositivos candidatos. *Continua* adoptou

normas de conectividade comprovadas, conforme é apresentado na Figura 2.9.

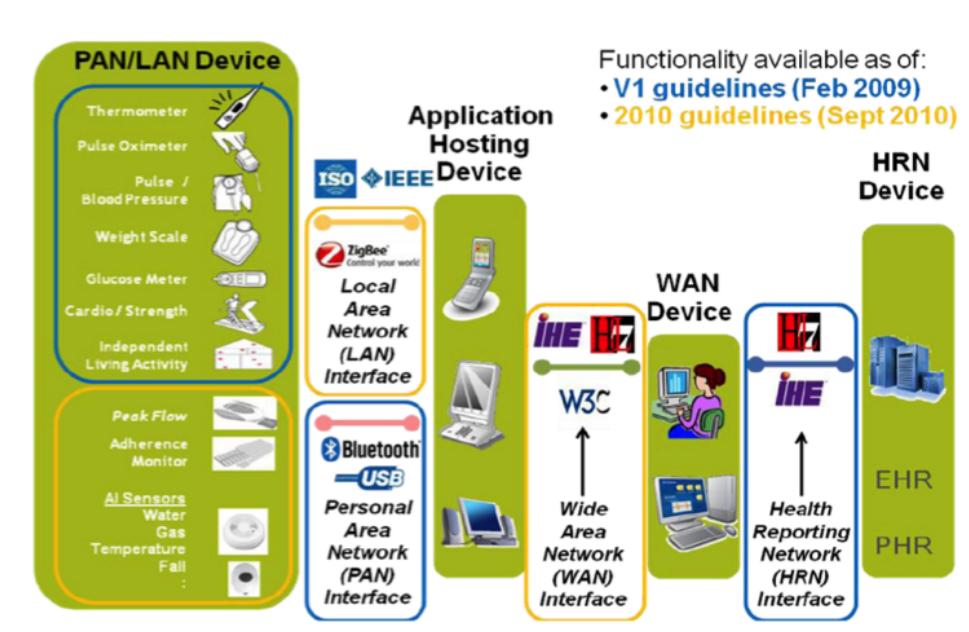


Figura 2.9: Interfaces e *Standards* da Continua

O programa de certificação da Continua [All13a] oferece um alto nível de certeza de que um dispositivo que possui o logotipo Continua foi projetado para estar de acordo com requisitos das *Design Guidelines da Continua* e com os padrões subjacentes. O processo descrito no web site da Continua [All13b] inclui quatro etapas, representadas na Figura 2.10, e que são descritas na Tabela 2.4.



Figura 2.10: Etapas do Processo de Certificação da Continua

O processo de certificação consiste no trabalho conjunto do fornecedor do dispositivo, do administrador de certificação e do laboratório de teste para documentar e verificar que um produto

Tabela 2.4: Descrição das Etapas do Processo de Certificação da *Continua*

Preparar	<ul style="list-style-type: none"> • Concluir o desenvolvimento do produto; • Estudar o processo de certificação e definir o tipo de certificação necessária. As diretrizes de certificação não são acessíveis gratuitamente. Um fornecedor tem de ser um membro da organização para obter essa documentação; • Selecionar o tipo de certificação necessária: <ul style="list-style-type: none"> – A para novos produtos; – D para produtos derivados de produtos existentes certificados com o tipo A; – O para produtos com certificação tipo A vendidos por um fornecedor OEM. • Começar os pré-testes, incluindo dois processos: <ul style="list-style-type: none"> – Conformidade de (pré)testes, utilizando ferramentas de teste disponíveis para os membros da Continua. – Testes de interoperabilidade (limitado a <i>PAN Devices</i>): documentação disponível apenas para membros.
Candidatar-se	<ul style="list-style-type: none"> • Aos membros é sugerido que apresentem o seu pedido, quando julgarem estar preparados para o teste. O membro é, portanto, obrigado a preencher o formulário de pedido de certificação; • É selecionado o laboratório de testes;
Testar	<ul style="list-style-type: none"> • O membro tem de enviar o seu dispositivo para o Laboratório de Testes pelo menos três dias antes da data do teste; • É recomendado que o membro assista ao teste; • O laboratório de teste envia os resultados do teste para a <i>Continua</i>.
Certificar	<p>Ao dispositivo é (em caso de sucesso) concedido o direito de utilizar o logótipo <i>Continua</i> depois de:</p> <ul style="list-style-type: none"> • Assinar o Contrato de Licença de Certificação da marca; • Assinar o Contrato de Licença de Dispositivo de Referência; • Pagar as taxas aplicáveis;

cumpre os requisitos para a certificação descritos numa especificação de certificação. Os membros da *Continua* devem seguir o processo da Figura 2.11 para certificar um dispositivo.

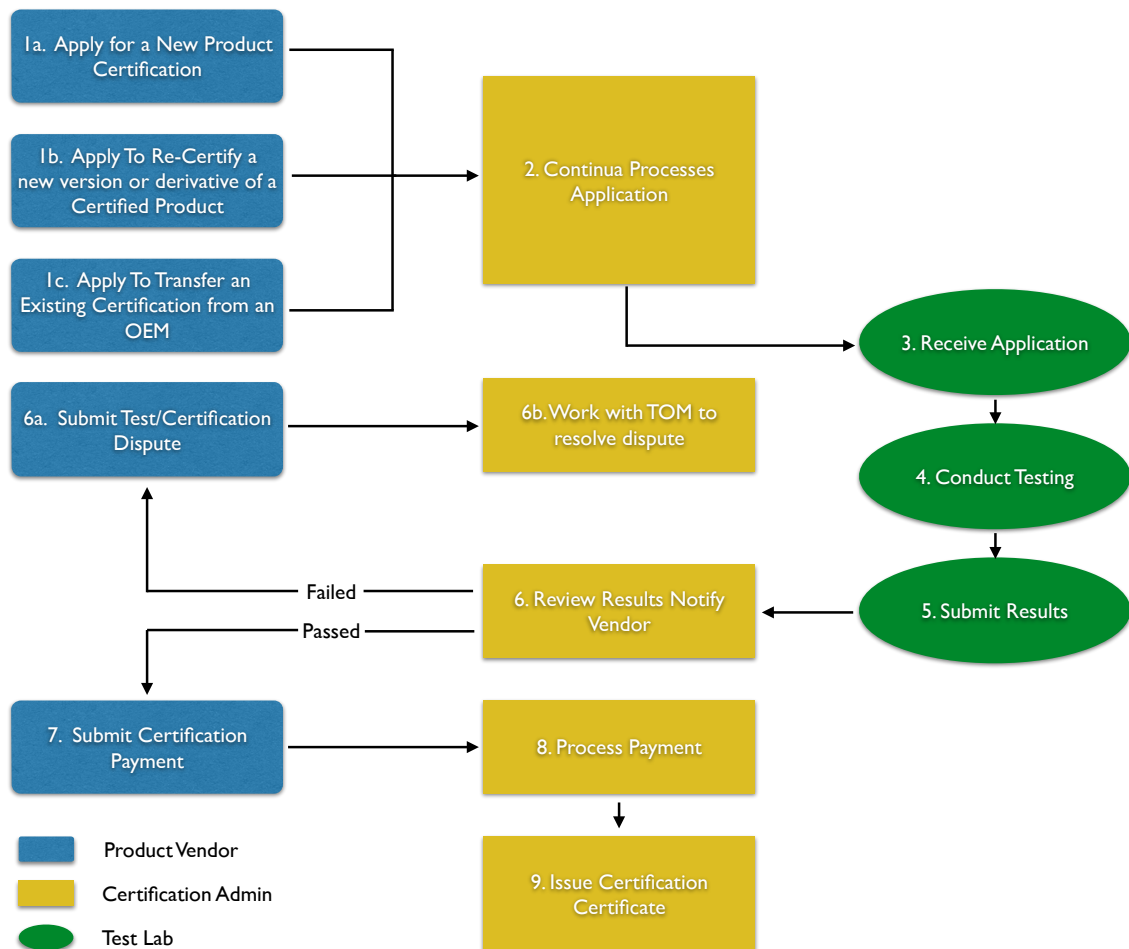


Figura 2.11: Workflow da Certificação da *Continua*

2.4.2 EHR-Q

EHR-Q [Eur13] é um projeto que pretende preparar melhor a comunidade de saúde em toda a Europa para a garantia da qualidade sistemática e comparável e certificação de produtos de *eHealth*, mais especificamente na área dos sistemas de registo electrónico de saúde (*Electronic Health Record* - EHR). Os beneficiários deste projeto são os países representados na Figura 2.12.



Figura 2.12: Países Benefeciários do Projeto EHR-Q

Relativamente à importância da rotulagem de qualidade e certificação, o projeto EHR-Q enumera cinco razões principais que podem levar um país a adotar a Certificação EHR-Q, sendo elas:

- Assegurar o cumprimento de normas e padrões nacionais;
- Aumentar a qualidade dos produtos através de uma funcionalidade coerente e pré-testada;
- Fomentar a troca de dados relativos aos cuidados de saúde e garantir a interoperabilidade dos sistemas relacionados;
- Melhorar a segurança do paciente quando recebe cuidados de saúde;
- Ter uma fonte de dados confiável para uso secundário.

A estratégia de certificação proposta pelo projeto EHR-Q distingue duas formas principais para rotulagem de qualidade e certificação. A primeira é uma abordagem do tipo *top/down* em que o público alvo e as autoridades são impostas por leis ou regulamentos. Esta abordagem é essencialmente nacional, uma vez que não existe uma entidade reguladora europeia para estas questões. A segunda é uma abordagem baseada no mercado, que tem por base a iniciativa de um fornecedor que pede um rótulo de qualidade a um prestador de serviços, sendo essa necessidade de estar em conformidade com uma série de critérios, uma parte de um processo de adjudicação. De forma a que os países, consigam internamente implementar o processo de certificação proposto, o

projeto EHR-Q produziu um roteiro padrão, que deverá ser seguido. Esse roteiro está representado na Figura 2.13.



Figura 2.13: Roteiro EHR-Q para a Certificação a Nível Nacional

Para além da certificação a nível interno, o projeto prevê uma certificação europeia e para isso produziu também um roteiro, que deve ser seguido pelos países interessados em implementar este tipo de certificação. Esse roteiro está representado na Figura 2.14.

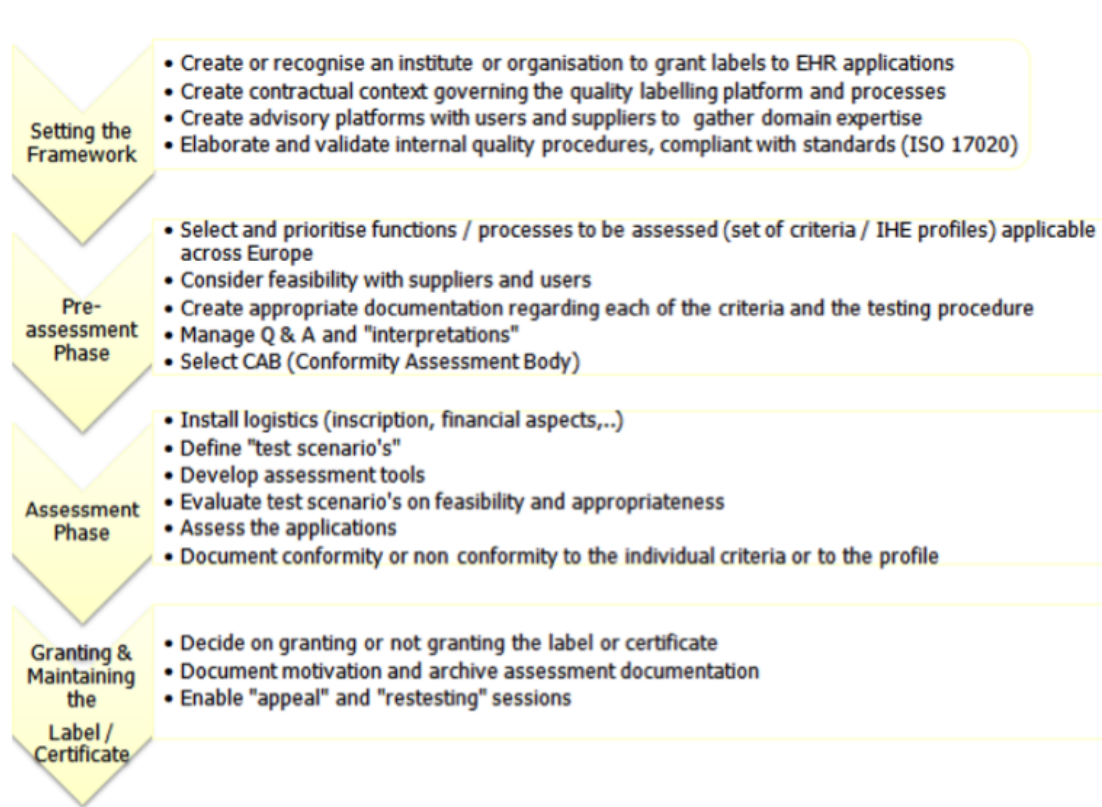


Figura 2.14: Roteiro EHR-Q para a Certificação a Nível Europeu

O projeto EHR-Q produziu ainda um conjunto de recomendações em diferentes áreas sobre o processo de rotulagem e certificação que idealizou. De entre as diversas recomendações, aquelas que podem ser úteis para este projeto de dissertação, estão representadas na Tabela 2.5.

Tabela 2.5: Recomendações do Projeto EHR-Q

<p>Em relação à rotulagem de qualidade e ao processo de certificação</p>	<ul style="list-style-type: none"> • Avaliação de terceiros é o procedimento mais adequado para o mercado ainda imaturo dos sistemas EHR, sendo a que oferece a maior garantia para os “clientes”. • Abordagens mistas estão a ser utilizadas, nomeadamente a auto-avaliação de uma série de declarações e avaliações de terceiros. Este é o caso na Bélgica, onde os critérios previamente testados são auto-avaliados pelo fornecedor, mas ainda objeto de ensaios aleatórios de validação. • É altamente recomendável iniciar pelo “pequeno”, para avaliar a eficácia do regime em vigor e as partes envolvidas, aumentando o foco e os requisitos a serem validados, passo a passo. • O modelo incentivado parece de longe o modelo mais promissor para promover um mais generalizado uso dos sistemas certificados, pelo menos para profissionais autónomos na área da saúde. O modelo obrigatório pode funcionar em alguns países com profissionais que só trabalham na área da saúde ou apenas com novas pessoas empregadas nesta área da saúde, embora para isso possa ser necessário um incentivo para criar estes empregos.
<p>Em relação aos aspetos da certificação a nível europeu</p>	<ul style="list-style-type: none"> • Rótulos de qualidade e certificados são essencialmente nacionais. É necessário fortalecer a certificação nacional de modo a melhorar a qualidade média e permitir um segundo passo, como a comparabilidade trans-Europeia e a portabilidade. • Os perfis IHE (Integrating the Healthcare Enterprise) são, em princípio independentes do país e devem ser promovidos como perfis europeus para troca de dados. • Promover a equivalência de certificados em toda a Europa através da validação a nível europeu das declarações funcionais descritas pelo EuroRec¹ e pelos perfis IHE. • Aplicar a portabilidade de rótulos de qualidade de avaliação e certificados em toda a União Europeia • Considerar a possibilidade de criar um “Registo de Qualidade Rotulada ou Certificado de Software Clínico” transfronteiriço, oferecendo informações sobre os produtos (sistemas de EHR, etc.) e documentação sobre o processo de certificação.

¹EuroRec é uma organização independente sem fins lucrativos, que promove na Europa o uso de sistemas de registo eletrónico de saúde de qualidade elevada.

2.4.3 HITCH

O projeto *The Healthcare Interoperability Testing and Conformance Harmonisation* (HITCH) produziu recomendações importantes para a Comissão Europeia sobre como proceder com testes de interoperabilidade de eHealth e certificação na Europa. HITCH [HIT13, Mic11] propõe um caminho a seguir equilibrando a urgência para desfragmentar a atual abordagem por projetos de eHealth, reduzir o custo dos testes e elevar o nível de qualidade, de modo a que seja suficientemente flexível para levar em conta as necessidades locais. HITCH considera testes e processos de certificação relacionados, mas distintos. Embora o teste represente a forma como se avalia/mede o nível de interoperabilidade, a certificação é um processo onde uma entidade certificadora utiliza os resultados de testes positivos. Se as especificações e as ferramentas de teste são robustas, a certificação torna-se uma mera formalidade já que um alto nível de interoperabilidade eHealth é realizado. HITCH [HIT13, Mic11] identificou duas áreas que precisam de melhorias e formalismo. Em primeiro lugar, existe uma necessidade de um guia de qualidade largamente aceite para o teste de interoperabilidade com base em normas de qualidade existentes, tais como a norma ISO 9001 [ISO08] e ISO 17025 [ISO05b]. Em segundo lugar, é necessária uma colaboração organizada para reduzir a fragmentação e falta de maturidade em ferramentas de interoperabilidade, de testes e planos de teste em áreas específicas. Na área de certificação, HITCH pesquisou estratégias de certificação diferentes e concluiu que diferentes variantes precisam de conviver e evoluir com estabilidade dos perfis de especificação e maturidade do mercado. A figura abaixo (Figura 2.15) mostra o conceito geral do projeto HITCH:

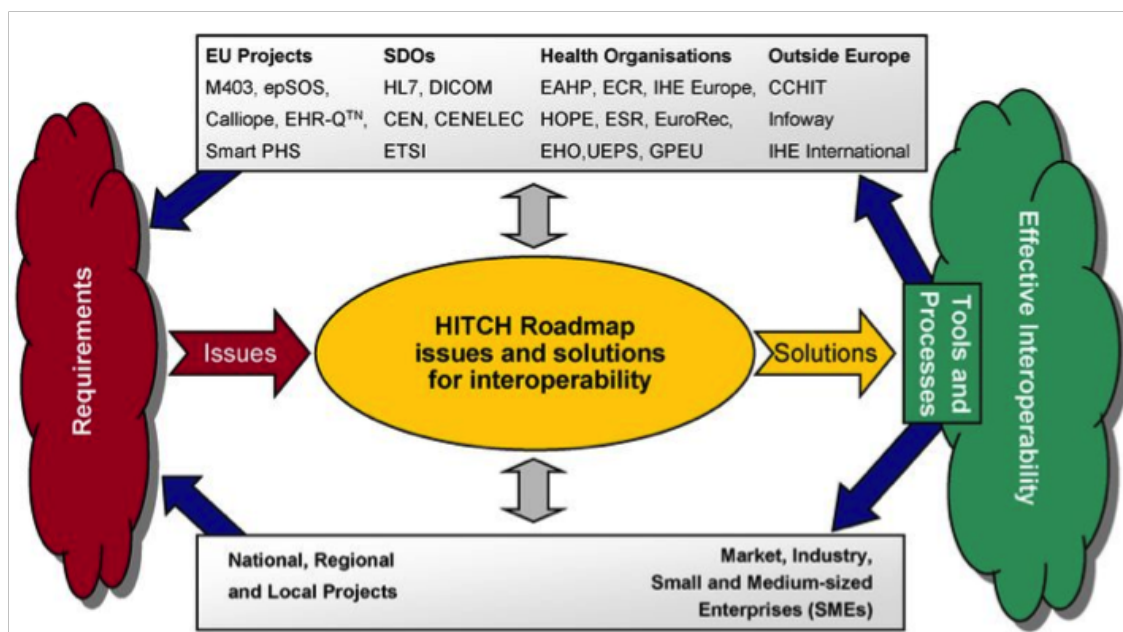


Figura 2.15: Conceito do projeto HITCH

O processo de certificação/rotulagem [Mic11] do projeto HITCH segue o processo da Figura 2.16.

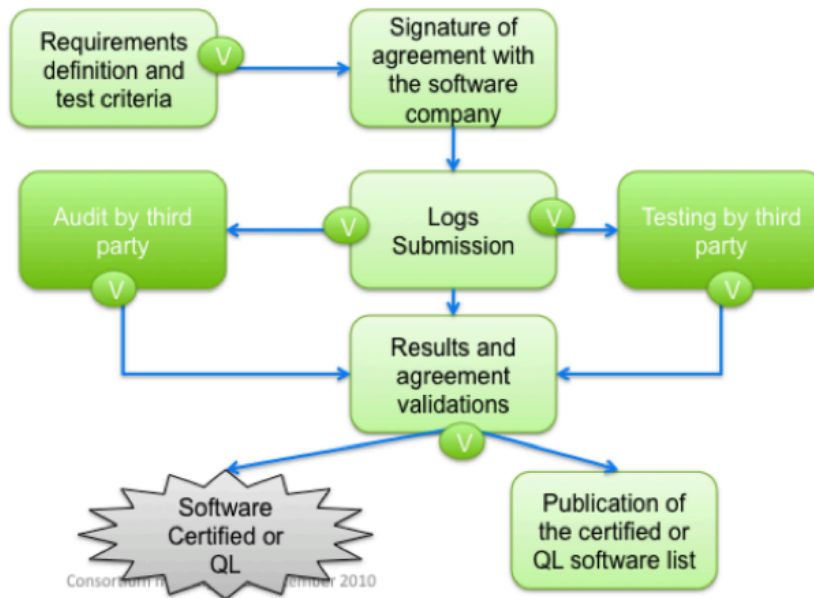


Figura 2.16: Processo de certificação/rotulagem do projeto HITCH

Todos os cenários de certificação [Mic11] definidos pelo projeto HITCH são baseados neste processo que descreve as atividades que um prestador tem que seguir a fim de apresentar produtos ou serviços para certificação. Os requisitos, especificações e critérios de teste são fornecidos pela organização que criou um selo/certificado para produtos e serviços. O fornecedor assina um acordo com esta organização, ou por delegação ao *Conformity Assessment Body* (CAB). Uma vez que o processo é iniciado, o fornecedor envia log's (resultados de testes e / ou documentação) para o CAB. Após a validação, uma inspeção ou um teste de laboratório é iniciado. Após a validação, o rótulo/ certificado é dado ao produto ou ao serviço testado. O projeto HITCH segue o conceito definido pela norma ISO 17011 [ISO04b] e dá uma visão abrangente das relações entre as diferentes entidades envolvidas no processo de certificação (Figura 2.17).

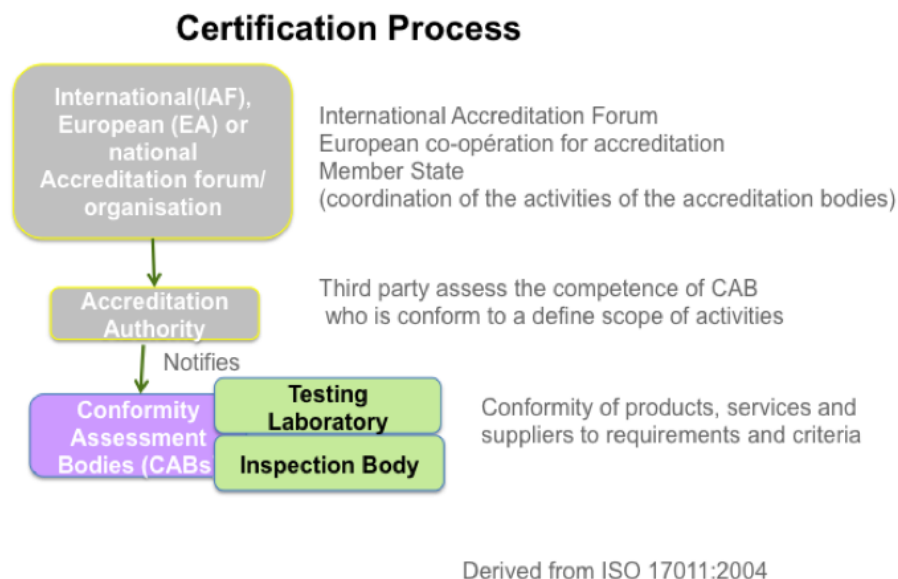


Figura 2.17: Relações entre as entidades envolvidas no processo de certificação

A autoridade de acreditação é uma entidade europeia/nacional que está em conformidade com a norma ISO 17011 [ISO04b] e/ou guia 65 [ISO96]. Esta entidade pode ser membro de uma rede de organismos de certificação onde as atividades de tais entidades são coordenadas, podendo ser a nível europeu ou internacional. Isto dá-lhes reconhecimento ao nível da sua competência, permitindo que o certificado dado a produtos por uma dessas entidades possa ser reconhecido em outros países. O projeto HITCH introduz, portanto, uma rotulagem estruturada e um processo de certificação para o domínio da eHealth. Considerando que o processo de rotulagem e o processo de certificação são muito semelhantes e que a diferença é mais uma questão de confiança dos resultados, o projeto HITCH decidiu fundir os dois processos num só, chamado LabelCert [Mic11].

A distinção entre os cenários LabelCert (usando o processo de LabelCert) baseia-se nos seguintes princípios:

- Entidades participantes para o processo de LabelCert: CAB (para certificação), organismo de controlo ou laboratório do organismo de testes e os participantes que apresentam produtos e os seus próprios processos de testes.
- O âmbito das atividades dos cenários LabelCert são a auditoria do processo de teste e/ou o teste de produtos que abrange dois aspectos: conformidade e testes de interoperabilidade cruzada (dois ou mais sistemas de teste de interoperabilidade entre eles).

Na figura seguinte (Figura 2.18) são apresentados diferentes cenários que correspondem a uma combinação de diferentes organismos e atividades.

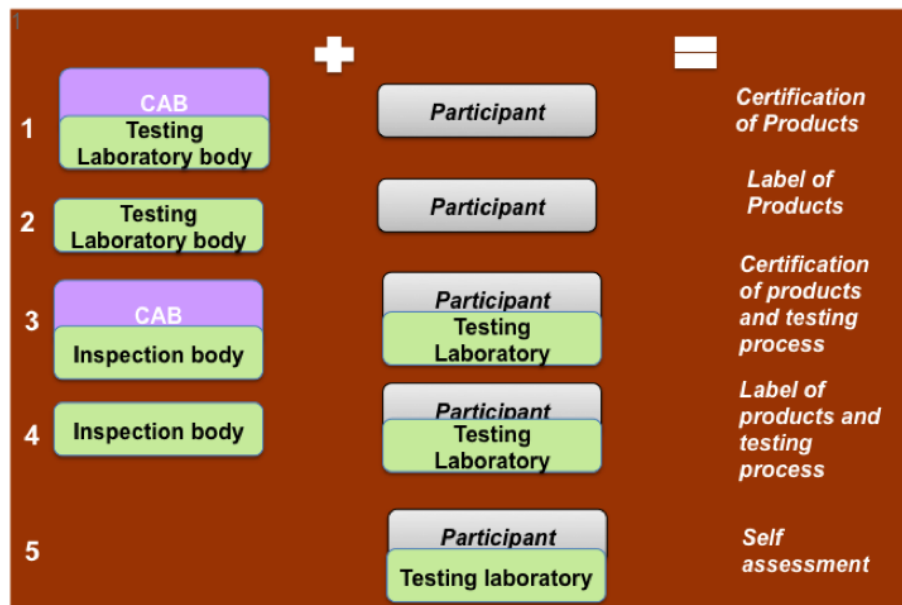


Figura 2.18: LabelCert - Cenários típicos

Os três cenários LabelCert [Mic11] que foram definidos pelo Projeto HITCH são:

- **Certificação de produtos (certificação por terceiros):** O processo de testes é realizada por um organismo de avaliação da conformidade (terceiro credenciado), de acordo com as especificações e requisitos que estão no âmbito da certificação (Correspondente ao cenário 1 na figura anterior).
- **Rótulo dos produtos e dos processos de ensaio (Auto-avaliação com referência externa):** O processo de teste é realizado pelo fornecedor (como auto-avaliação), de acordo com as especificações e os requisitos (que representa o âmbito da etiqueta) e sob uma qualidade de interoperabilidade aprovada pelo quadro de garantia. O fornecedor está sujeito a uma auditoria externa realizada por um organismo de inspeção (de terceiros).
- **A auto-avaliação de produtos:** O processo de teste é realizado pelo fornecedor de acordo com as especificações de ensaio e requisitos (que representam a atividade) e sob um quadro de garantia acordado de interoperabilidade e qualidade.

Os três projectos na área da saúde apresentados, contribuem cada um com recomendações que são úteis neste projeto. Obviamente a iniciativa *Continua* é aquela que mais pode ser útil, visto que é um projeto semelhante, embora não totalmente igual ao AAL4ALL, uma vez que não incorpora um modelo de negócio, uma infraestrutura de integração, nem é uma iniciativa aberta. No entanto o processo de certificação da *Continua* é uma boa base de partida para a metodologia de teste e certificação que se pretende definir neste projeto.

Revisão Bibliográfica

Capítulo 3

Proposta de Metodologia de Teste e Certificação

Ao longo deste capítulo é descrita de forma detalhada a proposta de metodologia de teste e certificação desenvolvida. Para isso são abordados os quatro principais aspetos, processo de teste e certificação, metamodelos e *templates*, estratégia de teste e certificação e por último os certificados. Uma versão anterior e menos detalhada da metodologia proposta pode ser encontrada no artigo *A Testing and Certification Methodology for an Ambient-Assisted Living Ecosystem* [FLBM13] que foi apresentado na *IEEE HealthCom'13*¹.

3.1 Processo de Teste e Certificação

A proposta de metodologia de teste e certificação elaborada, tem por base o modelo de negócio definido para o projeto AAL4ALL [AAL13], modelo esse que é acente em duas entidades distintas que possuem responsabilidades e missões diferentes.

A primeira entidade é a designada por Associação AAL4ALL, que assume o papel de entidade normalizadora e dinamizadora do AAL4ALL. Esta associação é composta pelos parceiros do AAL4ALL, entidades do HCP interessadas na temática AAL, bem como outras entidades que possam estar interessadas ou possuir alguma ligação à temática AAL. As responsabilidades desta associação (representadas de forma esquematizada na Figura 3.1) são:

- Promoção de actividades, produtos e serviços dos seus associados;
- Definição, actualização e promoção do referencial AAL4ALL;
- Contribuição para o desenvolvimento de novas normas;
- Acreditação/Reconhecimento das entidades que pretendam integrar a entidade certificadora;

¹IEEE HealthCom'13 é uma conferência internacional na área das tecnologias para a saúde e AAL

Proposta de Metodologia de Teste e Certificação

- Dinamização de *networking* e outras actividades de interesse comum dos associados;
- Papel de “Provedor do cliente” nas temáticas AAL para o utilizador final.

A sustentabilidade desta associação é assegurada não só através de uma taxa cobrada a cada associado, mas também através de receita decorrente da organização de eventos temáticos e da participação em projetos co-financiados de AAL.



Figura 3.1: Responsabilidades da Associação AAL4ALL

A segunda entidade é denominada por Entidade Certificadora AAL4ALL, e assume o papel de entidade certificadora e comercializadora do rótulo AAL4ALL. A entidade certificadora pode assumir a forma de uma associação e nesse caso subcontrata à sua rede de parceiros (seus associados) testes e ensaios de validação necessários ao processo de certificação. Esta associação é composta por principalmente dois grupos de associados, parceiros AAL4ALL e outras instituições, desde que não possuam ou pretendam vir a possuir produtos e/ou serviços no mercado que possam vir a ser potencialmente certificáveis, salvaguardando deste modo que não existe um conflito de interesses. Organizações que pretendam fazer parte do processo de certificação de produtos AAL4ALL e que disponham de meios e competências para a realização de testes e ensaios de validação necessários à certificação, sendo que também estas entidades, não podem possuir produtos e/ou serviços potencialmente certificáveis para mais uma vez garantir que não existe um conflito de interesses. Todas as adesões a esta entidade são ainda ratificadas pela Associação AAL4ALL que terá um

papel de acreditação sobre a adequação dos meios e/ou competências para a realização dos testes e ensaios de validação.

As responsabilidades desta entidade (representadas de força esquematizada na Figura 3.2) são:

- Definição dos procedimentos de teste e ensaio, assegurando a implementação do modelo de certificação;
- Realização de auditorias;
- Realização de testes e ensaios (subcontratando os seus membros para o efeito);
- Comercialização do rótulo AAL4ALL (Produto Certificado);

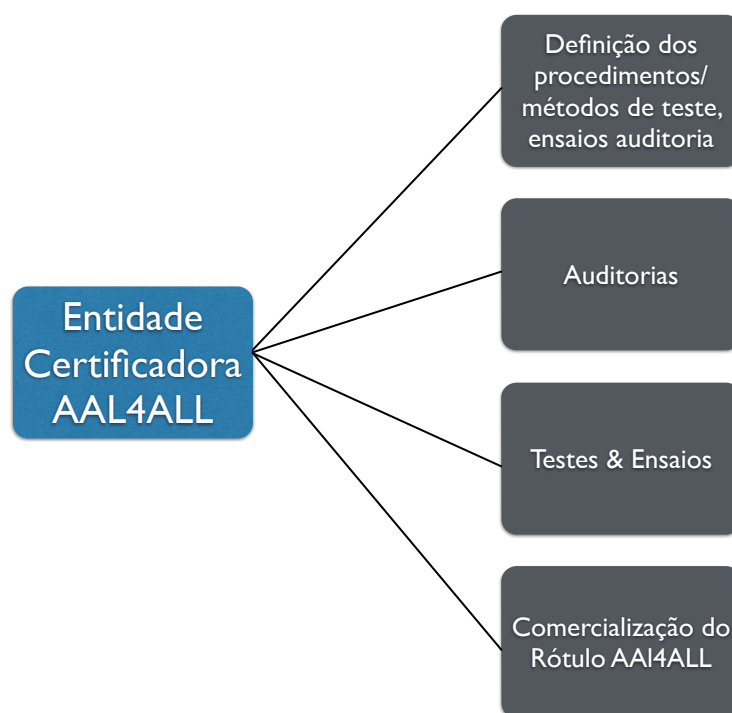


Figura 3.2: Responsabilidades da Entidade Certificadora AAL4ALL

Esta será uma entidade auto sustentável, que gerará resultados financeiros principalmente através da cobrança de uma taxa associada a cada processo de certificação, mas também através da cobrança de uma taxa à Associação AAL4ALL sempre que existir a necessidade de introduzir ou atualizar uma norma (que implique uma alteração dos processos de teste).

Tendo em conta as duas entidades descritas anteriormente, bem como as suas responsabilidades, o processo de teste e certificação elaborado iniciar-se-á sempre com o pedido de um fabricante de um determinado produto para o qual pretende obter o rótulo AAL4ALL. O processo de teste e certificação genérico será então o representado na Figura 3.3. No entanto também existem casos

específicos que não seguem o processo genérico, nomeadamente quando se trata de uma renovação, ou mesmo quando um fabricante pretende alterar algo no seu produto certificado (mudança de funcionalidades por exemplo). Nestes casos o processo será mais simples e rápido, sem que no entanto isso signifique menos rigor na avaliação.

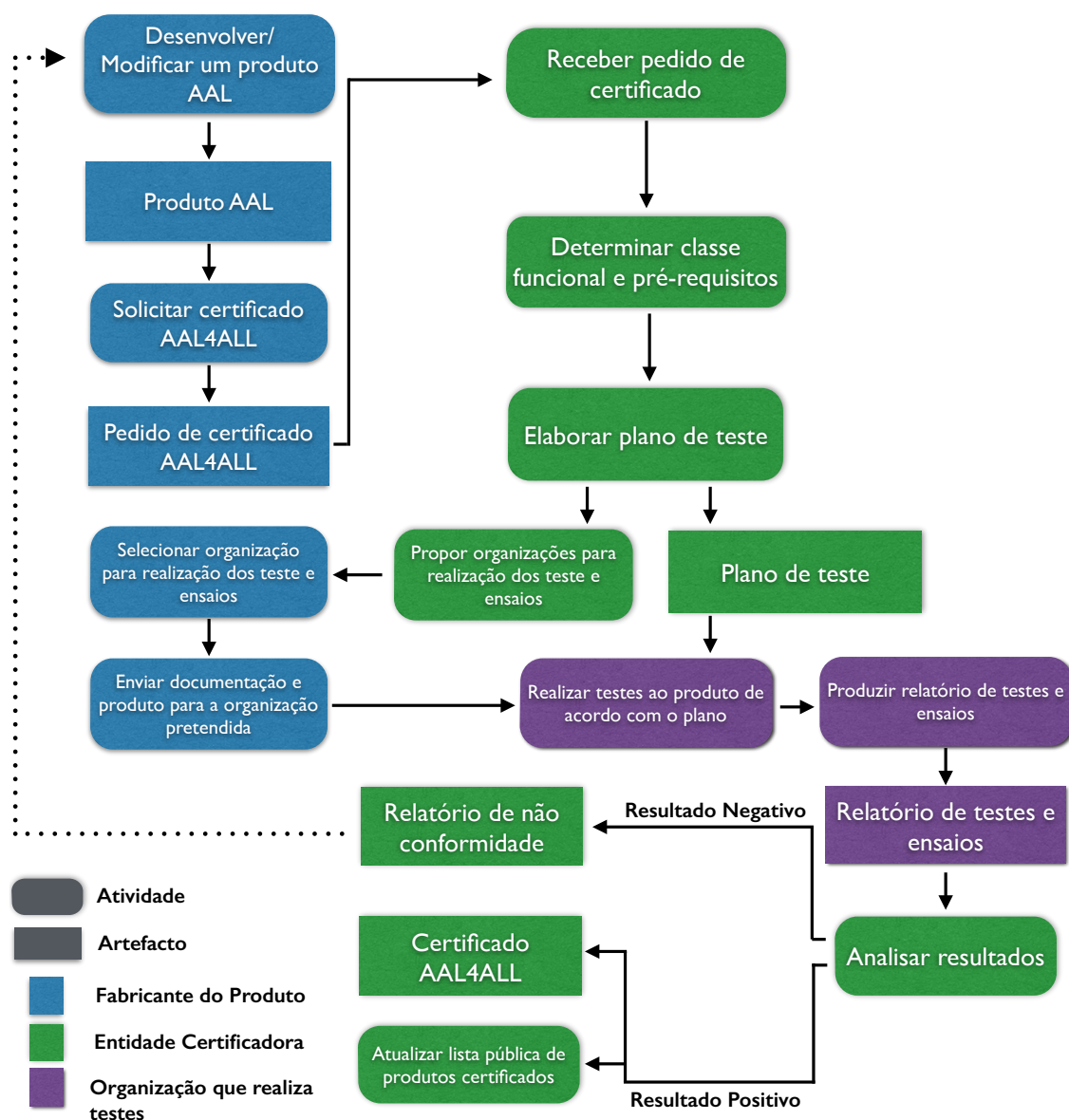


Figura 3.3: Processo de Certificação AAL4ALL

Os intervenientes referidos anteriormente, nomeadamente a associação AAL4ALL, a entidade certificadora e as organizações que realizam teste e ensaios devem estar sujeitos aos requisitos definidos pelas normas ISO/IEC 17011:2004, ISO/IEC Guia 65:1996 e ISO/IEC 17025:2005 respetivamente (ver Secção 2.1.1).

O processo de certificação AAL4ALL foi inspirado no processo de certificação da *Continua*

(ver Secção 2.4.1), tendo sido adaptado ao modelo de negócio do projeto AAL4ALL. Este processo foi ainda elaborado tendo em conta a norma ISO/IEC Guia 28:2004 (ver Secção 2.1.1).

3.2 Metamodelos e Templates

A Figura 3.4 apresenta uma visão geral do metamodelo proposto para a especificação de produtos e serviços AAL normalizados (denominadas *ComponentClass* e *ServiceClass* respectivamente) e os casos de testes associados.

Neste metamodelo, um serviço deve ser entendido como um serviço de negócio ponta a ponta, que envolve um conjunto de atores (receptores e prestadores de cuidados de saúde) e componentes (dispositivos físicos ou aplicações de software), cuja interação é descrita através de diagramas de sequência UML [Gro11] (um exemplo de um desses diagramas pode ser observado na Figura 3.5). Este tipo de serviço não deve ser confundido com um serviço de software.

Para cada produto AAL devem ser descritas as suas interfaces (interfaces humanas ou interfaces de software), mensagens que são capazes de enviar e receber através dessas interfaces, os eventos que desencadeiam o envio de mensagens (e.g. receber uma mensagem, atingir um estado interno, etc.), os efeitos das mensagens recebidas (e.g. alterar o estado interno ou enviar uma mensagem), as condições em que as mensagens podem ser recebidas (e.g. estar em algum estado interno) e pré-requisitos que devem satisfazer (e.g. rotulagem CE [Com13] ou conformidade com a norma ISO/IEEE 11073 [ISO04f]). Um exemplo de como pode ser descrito um produto tendo em conta todos estes aspetos, pode ser observado na Tabela 3.1.

Uma vez que o projeto AAL4ALL já possuía um *template* de documento para descrição de produtos, foi realizada uma análise a esse mesmo *template*, tendo essa análise revelado que o *template* utilizado até então para a descrição de produtos possuía lacunas, ao nível da descrição das interfaces dos componentes, lacunas essas que colocavam em causa o processo de certificação. De forma a colmatar essas lacunas foi proposto ao projeto AAL4ALL a alteração do *template*, nomeadamente a incorporação de secções específicas para as descrições das interfaces dos componentes candidatos a certificação tendo por base o exemplo descrito na Tabela 3.1. O *template* resultante desta adaptação pode ser encontrado no Anexo C.

Relativamente às especificações dos testes de conformidade e dos testes de interoperabilidade, foi também necessário definir um *standard*, por forma a uniformizar esse processo. Para isso foi adoptada a norma IEEE Std 829TM-2008 [IEE08]. Um exemplo da descrição de um teste de conformidade segundo esta norma pode ser observado na Tabela 3.2

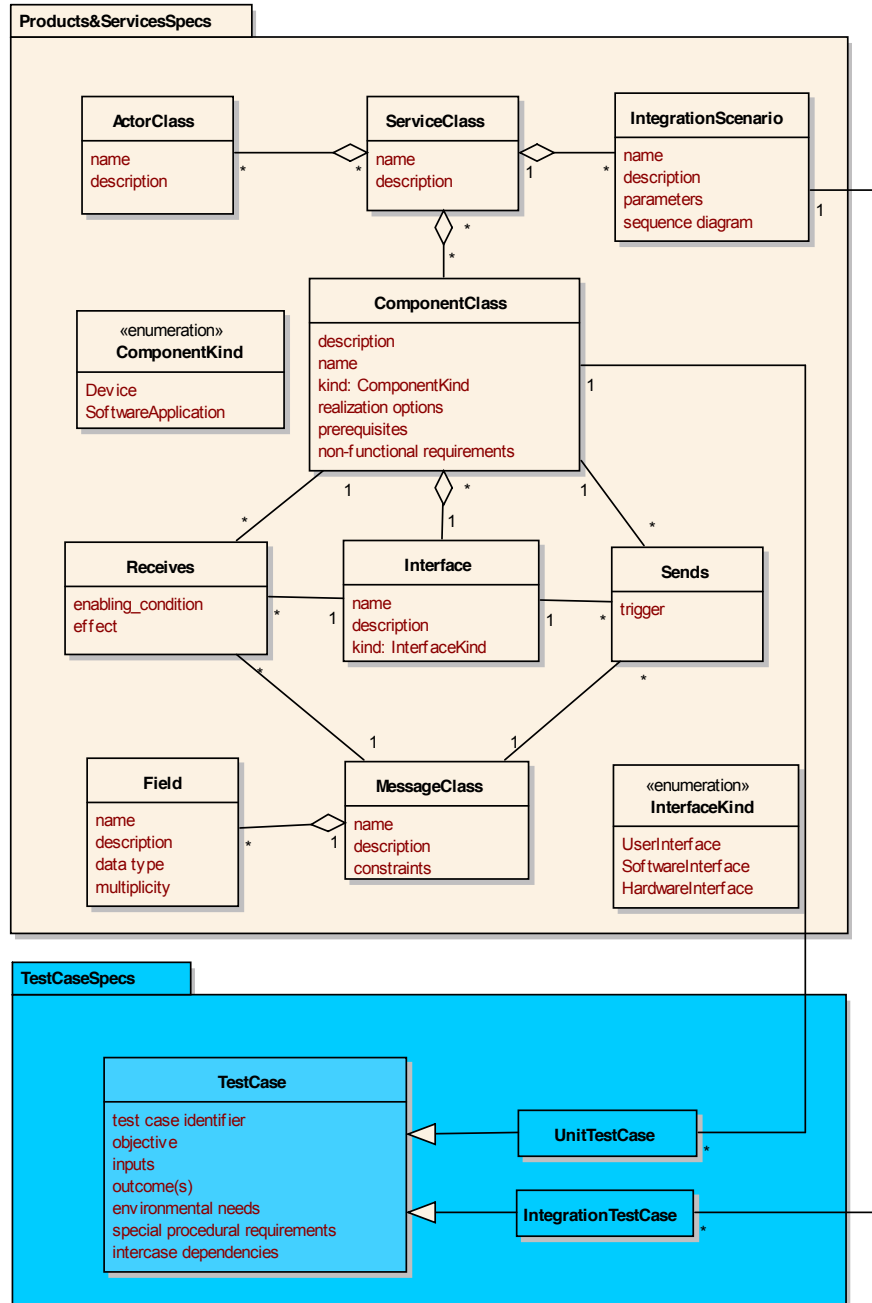


Figura 3.4: Diagrama de classes UML do metamodelo para a especificação de produtos e serviços AAL normalizados e os casos de teste associados

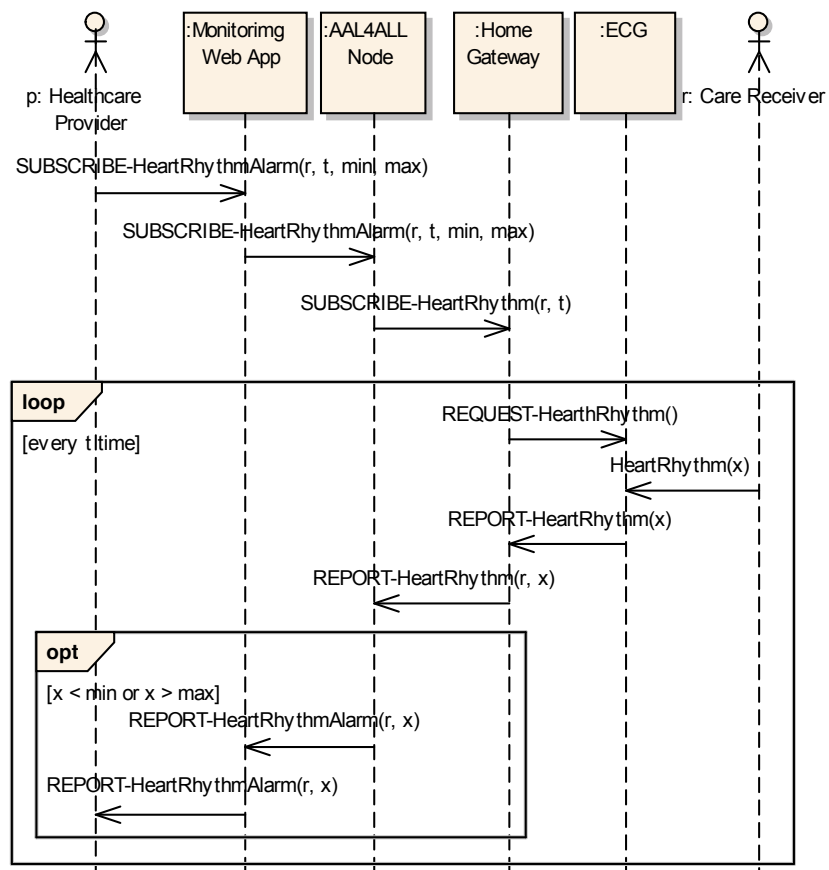


Figura 3.5: Diagrama de sequência UML para descrição das mensagens enviadas/recebidas pelos componentes de um determinado cenário de integração

Tabela 3.1: Exemplo da descrição de uma categoria de componente

Name	Home Gateway
Description	Manages communication with in-home devices (sensors and actuators) and AAL4ALL Nodes.
Kind	Device
Interfaces	name: I1 description: Interface with AAL4ALL Node. kind: SoftwareInterface
	name: I2 description: Interface with sensors and actuators. kind: HardwareInterface.
State variables	Table of subscriptions
	Table of sensors and actuators
Receives	message: SUBSCRIBE-HeartRhythm(r, t) interface: I1 enabl.cond.: always enabled effect: every t time do send REQUEST-HeartRhythm() to ECG of Care Receiver r
	message: REPORT-HearthRythm(x) interface: I2 enabl.cond.: a REQUEST was sent effect: send REPORT-HearthRythm(r, x), where r is the Care Receiver associated with the ECG sensor
Sends	message: REQUEST-HeartRhythm() interface: I2 trigger: see SUBSCRIBE-HeartRhythm(r, t)
	message: REPORT-HearthRythm(r, x) interface: I1 trigger: see REPORT-HearthRythm(x)
Realization options	Communication protocols at I2: Wi-Fi, Bluetooth, Ethernet
Prerequisites	CE marking
Non-funct. requirements	Message sending delay is less than 5 seconds.

Tabela 3.2: Descrição de um teste de conformidade segundo a norma IEEE Std 829™-2008

Test case identifier	HG.1
Objective	Check if a home gateway, when requested to provide HearthRhythm information about a specified user r with a specified periodicity t , sends requests for HearthRhythm information to the appropriate user equipment with the appropriate timing.
Inputs	Configuration information mapping user identifiers and equipment identifiers. SUBSCRIBE-HeartRhythm(r, t)
Outcome(s)	REQUEST-HeartRhythm() messages, to the correct device, with t time period, with a maximum delay of 5 sec.
Environmental needs	Message injection and monitoring equipment.
Special procedural requirements	Before sending the SUBSCRIBE message, the gateway must be configured.
Requirement identifier	R1
Intercase dependencies	-

3.3 Estratégia de Teste

Tendo em conta o que já foi descrito anteriormente neste capítulo, foi elaborada uma estratégia de teste assente em três partes fundamentais que são descritas nas subsecções seguintes.

3.3.1 Fases do Processo de Teste

O processo de teste proposto tem como objetivo garantir de forma inequívoca que qualquer produto que o conclua com sucesso, está efetivamente apto a entrar no ecossistema de produtos e serviços AAL. Após uma análise aprofundada das várias soluções existentes e já descritas no capítulo 2, nomeadamente na secção 2.3, chegou-se à conclusão que o processo que cada componente candidato deveria ser sujeito deve ser composto por três passos, sendo eles:

1. Verificação do cumprimento dos pré-requisitos do componente candidato, através da análise de evidências apropriadas;
2. Execução de testes de conformidade derivados da especificação do componente candidato cobrindo todas as suas mensagens, bem como os seus efeitos, condições e *triggers*;
3. Execução de testes de interoperabilidade derivados dos cenários de integração dos serviços em que o componente candidato participa (utilizando para isso componentes certificados como restantes participantes), abrangendo todas as mensagens e suas derivações.

O primeiro passo consiste na verificação, através da análise de evidências, se o componente candidato cumpre ou não os pré-requisitos exigidos pela entidade certificadora para a categoria

de produto em que o mesmo se insere. Este tipo de pré-requisitos, pode ser desde a necessidade de possuir por exemplo a marcação CE [Com13] ou a necessidade de possuir uma norma de comunicação específica como é o caso da norma ISO/IEEE 11073 (ver Secção 2.2) adoptada hoje em dia pelos dispositivos médicos. Este tipo de análise é facilmente realizada através da observação de evidências, uma vez que é suficiente observar o componente, ou a sua documentação para se constatar se o mesmo cumpre ou não um determinado requisito. Por forma a facilitar a sua análise por parte da organização que executar os testes, os pré-requisitos devem estar descritos numa *check list* uma vez que só existem duas opções durante a sua análise, o componente cumpre ou não cumpre o pré-requisito em questão. Só um componente que concluir a *check list* de pré-requisitos com sucesso, ou seja cumpre efetivamente todos os pré-requisitos passará então ao segundo passo do processo de teste.

O segundo passo deste processo, consiste na submissão do componente candidato à realização de testes de conformidade, testes esses descritos pela entidade certificadora, que cobrem de forma completa as mensagens enviadas e recebidas por esse componente nas suas interfaces de comunicação, quer elas sejam interfaces de software ou interfaces humanas. Neste passo o objetivo é garantir que o componente candidato comunica de acordo com aquilo que está especificado nas normas, e documentação do produto para aspetos em aberto na própria norma, bem como se realmente proporciona ao utilizador (nos casos em que isto se aplica) aquilo a que se compromete apresentar. Embora não esteja proposto nesta metodologia de teste e certificação a realização de teste de usabilidade, se no futuro viessem a ser sugeridos e/ou considerados úteis por parte da entidade certificadora, os mesmo poderiam facilmente ser inseridos neste passo do processo de teste e certificação, uma vez que para isso bastaria, em vez de se verificar apenas se o componente candidato apresenta ao utilizador aquilo a que se compromete, poderia-se avaliar essa apresentação de acordo com determinados critérios que seriam descritos pela entidade certificadora. Neste passo os casos de teste, são descritos segundo a norma IEEE Std 829TM-2008 [IEE08], já anteriormente apresentada. O resultado possível para cada teste pode ser:

- *Passed* - Resultado atribuído a um teste que foi concluído com sucesso;
- *Inconclusive* - Resultado atribuído a um teste quando os resultados levantem dúvidas à organização que está a realizar os testes;
- *Not tested* - Resultado atribuído quando não é possível, por alguma razão executar um determinado teste. Isto pode acontecer por exemplo, no caso de a documentação do componente candidato não permitir que a organização reponsável pela execução dos teste perceba de que forma o pode executar;
- *Failed* - Resultado atribuído a um teste que falhou.

Só um componente que obtiver o resultado de *Passed* em todos os testes do segundo passo, passará ao terceiro e último passo do processo de teste. Sendo que os testes realizados no segundo passo que obtiveram resultados *Inconclusive*, *Not tested*, *Failed* são comunicados à entidade certificadora, com a respectiva justificação e/ou explicação da sua atribuição de forma a que esta possa

comunicar ao fabricante do componente o motivo que levou a que o componente não completasse o processo de teste e certificação.

O terceiro e último passo do processo de teste e certificação, consiste na submissão do componente candidato à realização de testes de interoperabilidade, testes esses que mais uma vez são descritos pela entidade certificadora. Estes testes cobrem os serviços nos quais o componente candidato irá participar, quando estiver no mercado, sendo por isso o objetivo deste passo, comprovar que o componente candidato se comporta como é esperado quando integrado com outros componentes. Por forma a executar estes testes a organização responsável irá sempre utilizar componentes já certificados como restantes elementos do cenário de integração, sendo que aquilo que será avaliado é a evidência entre o resultado esperado e o resultado obtido, sendo a análise feita apenas ponta a ponta, ou seja, através do papel dos atores envolvidos no cenário de integração. Neste tipo de testes a classificação obtida será, assim como no passo anterior uma de entre as quatro possibilidades, *Passed*, *Inconclusive*, *Not tested* ou *Failed*. Só um componente que obtenha o resultado *Passed* em todos os testes de interoperabilidade completará com sucesso o processo de teste e certificação e obterá então o certificado que lhe permitirá exibir o rótulo AAL4ALL.

3.3.2 Abordagem Incremental de Configurações de Teste

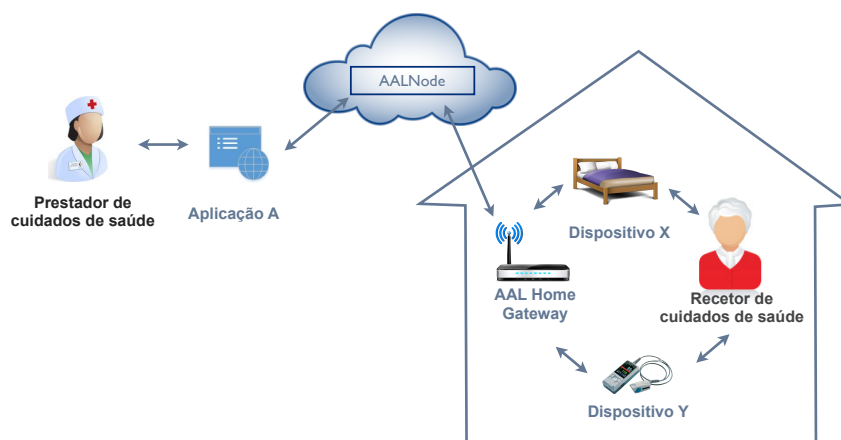


Figura 3.6: Cenário exemplo

A metodologia de testes e certificação proposta visa também a optimização do processo de teste e certificação. Desse modo é proposta uma abordagem incremental de configuração de teste, que permita não só iniciar a metodologia proposta, uma vez que no início não estão disponíveis produtos certificados que possam ser utilizados por exemplo no terceiro passo do processo de teste e certificação, mas também apresente uma solução para aquilo que pode ser automatizado neste processo. Com esse objetivo e utilizando um cenário exemplo (representado na Figura 3.6) e os seus componentes, é sugerido que o processo de teste e certificação se inicie sobre o componente AALNode, uma vez que é uma peça central deste cenário.

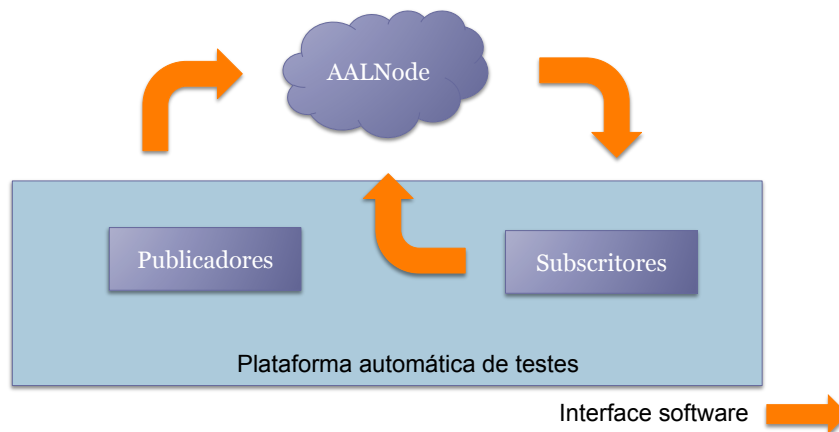


Figura 3.7: Interfaces do AALNode

Como representado na Figura 3.7, o componente AALNode possui três interfaces, todas elas interfaces de software, sendo por isso possível a criação de uma plataforma automática que permitirá o envio e recepção de mensagens nessas interfaces de forma a automatizar o processo de teste, nomeadamente dos testes de conformidade.

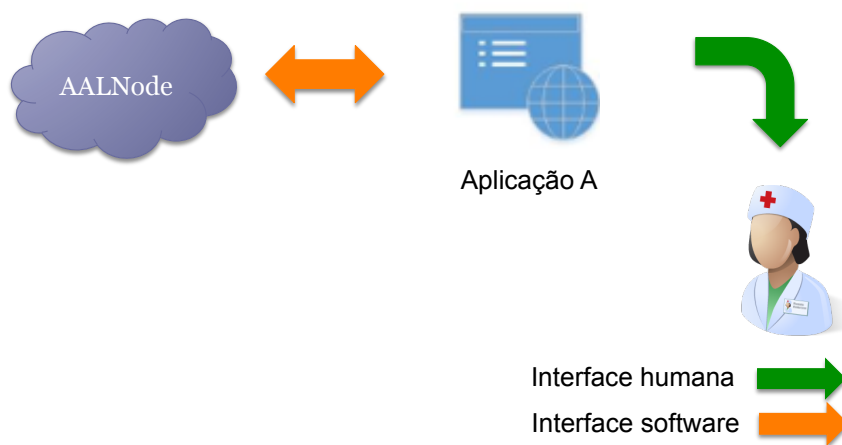


Figura 3.8: Interfaces da Aplicação A

Uma vez realizados com sucesso todos os testes de interoperabilidade sobre o AALNode, este será então o primeiro componente certificado com o rótulo AAL4ALL, sendo a partir deste momento possível, não só utilizar este componente para realizar testes de interoperabilidade, como também reaproveitar a plataforma automática criada para ele, de forma a serem realizados testes de conformidade sobre outros componentes que em algumas das suas interfaces consumam ou enviem mensagens do AALNode. Por exemplo, se quisermos agora realizar testes de conformidade sobre o componente “Aplicação A” e tendo em conta que este componente possui duas interfaces distintas (como representado na Figura 3.8) podemos utilizar a plataforma automática utilizada no

AALNode, para simular o envio e recepção de mensagens na interface de software da “Aplicação A”. A este processo de utilização de componentes certificados para certificar novos componente podemos chamar de abordagem incremental. Relativamente à outra interface da “Aplicação A”, a interface humana, não é possível automatizar os testes, uma vez que o que é avaliado neste caso são as evidências (e.g. mensagens) mostradas ao utilizador.

3.4 Certificados

Uma vez completado o processo de teste e certificação por parte de um componente candidato, esse componente está em condições de poder utilizar o rótulo “AAL4ALL”. No entanto existe ainda um último passo, para que o componente possa efetivamente poder utilizar esse rótulo. Este passo é o processo de atribuição do certificado. Uma vez obtido o resultado positivo no processo de certificação, o fabricante do componente é chamado à entidade certificadora, para oficializar a entrada deste produto na “Família de produtos AAL4ALL”, implicando isso a assinatura de um contrato onde, entre outras coisas, estão descritas as obrigações do fabricante, de entre as quais se destacam:

- Obrigação da utilização do rótulo no produto apenas na versão para a qual foi certificado, pois qualquer alteração efetuada ao produto, deve ser comunicada à entidade certificadora, uma vez que esta implica um novo processo de certificação;
- Obrigação do uso do rótulo “AAL4ALL” de acordo com as regras de imagem e comunicação estabelecidas pela entidade certificadora;
- Obrigação de comunicação à entidade certificadora de qualquer problema detetado com o produto e que possa colocar em causa os pressupostos que estiveram na base da atribuição do certificado.

Qualquer falha por parte do fabricante no cumprimento das suas obrigações, para além de eventuais processos judiciais, implicará a exclusão imediata do seu produto da lista publicada de produtos certificados, bem como passará a partir deste momento a ser proibido a comercialização do seu produto com o rótulo “AAL4ALL”. Por outro lado este contrato também contempla os direitos do fabricante do produto nomeadamente o compromisso de validade do certificado por um período de três anos e a integração do produto certificado no ecossistema de produtos e serviços da “Família de produtos AAL4ALL”.

Após a assinatura do contrato, o fabricante recebe da entidade certificadora, um diploma e as credências únicas do componente. O diploma é um documento, onde constam as seguintes informações:

- Identificação da entidade certificadora;
- Número do certificado;

Proposta de Metodologia de Teste e Certificação

- Identificação do fabricante do componente;
- Identificação do componente;
- Versão do componente;
- Identificação da organização responsável pela execução dos testes;
- Identificação da(s) categoria(s) para a(s) qual(ais) o produto está certificado;
- Data da emissão do certificado;
- Data de validade do certificado;
- Assinatura do presidente da entidade certificadora.

Um exemplo de um certificado pode ser observado na Figura 3.9. Relativamente às credenciais únicas do componente, estas servem para que o mesmo seja reconhecido pelo ecossistema AAL4AAL, permitindo dessa forma a sua comunicação com os restantes produtos da “Família de produtos AAL4ALL”.



PRODUCT CERTIFICATE

Number: 0000000XXXXX

Manufacturer: Manufacturer Y

Product: Aplication A

Version: 5.4

Product Classes
A, B and D

Test Laboratory
Organization x

Certification Date	Validity of certification
24/11/2013	24/11/2016



President of the certification authority

Figura 3.9: Exemplo de um certificado AAL4ALL

Proposta de Metodologia de Teste e Certificação

Capítulo 4

Aplicação e Validação da Metodologia

Neste capítulo é descrita a implementação da metodologia de teste e certificação, apresentada no capítulo anterior, no contexto real do projeto AAL4ALL.

4.1 Cenário de teste

Tendo em conta o estado atual do projeto AAL4AAL no que diz respeito à criação e documentação de componentes, bem como à disponibilidade das entidades envolvidas no projeto para cederem equipamentos que pudessem ser testados, o cenário de teste possível para validação da metodologia de teste e certificação, foi o representado na Figura 4.1. Este cenário é composto por uma aplicação denominada “Caretaker Application”, a aplicação “AALNode”, um sensor e uma *gateway* com um *driver* incorporado, representando um serviço de deteção de atividade.

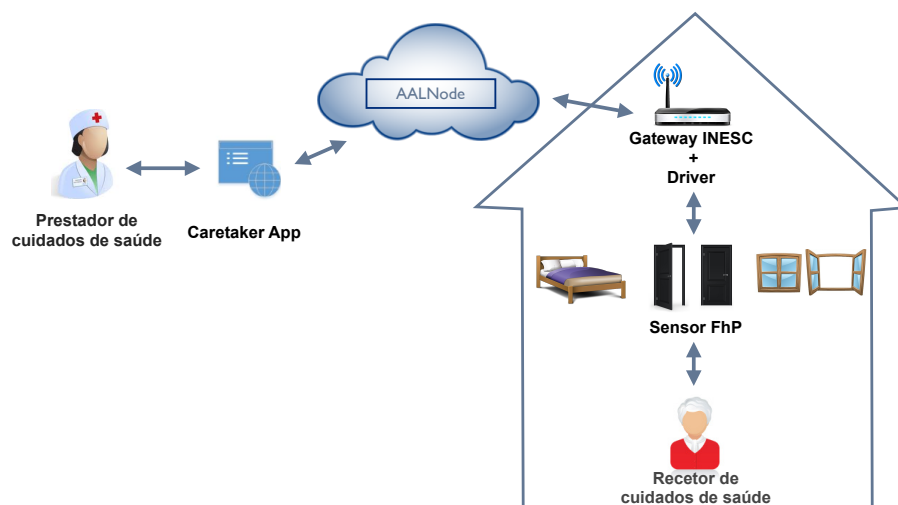


Figura 4.1: Cenário de teste

A aplicação “Caretaker Application”, desenvolvida pela Fraunhofer Portugal, é uma aplicação que permite que prestadores de cuidados de saúde tenham acesso a informações (dados de sensores, etc.) recolhidas na casa dos receptores de cuidados de saúde, sendo ainda capaz de interpretar esses mesmo dados para, por exemplo, gerar alertas quando determinados valores atinjam determinados patamares definidos previamente pelo prestador de cuidados de saúde responsável. No entanto, no momento da realização deste cenário esta aplicação encontrava-se na sua versão 0.5, onde as suas funcionalidades ainda se limitavam à receção de dados recolhidos numa determinada habitação e a demonstração dos mesmos ao recetor de cuidados de saúde.

O “AALNode”, desenvolvido pelo INESC TEC, é uma aplicação de software que funciona como um nó agregador de informação e que presta um serviço de *Bus* a todos os componentes do projeto AAL4ALL. O “AALNode” permite que subscritores se registem, para que a partir desse momento recebam mensagens de um determinado tópico sempre que um publicador as envie para o “AALNode”.

O sensor presente neste cenário e que foi desenvolvido pela Fraunhofer Portugal é capaz de ser aplicado em diferentes situações, tendo sido utilizado neste cenário com 3 funcionalidades diferentes. Foi utilizado como sensor de deteção da alteração do estado de uma porta (verificação se a mesma foi aberta ou fechada), alteração do estado de uma janela e ainda como sensor de deteção da alteração do estado numa cama, onde a análise é feita com recurso a uma banda deformadora que é colocada por baixo do colchão e que deteta se uma pessoa está ou não deitada na mesma.

A *gateway* utilizada neste cenário foi desenvolvida pelo INESC TEC e possuía um driver desenvolvido pela Fraunhofer Portugal que permite à *gateway* receber informação do sensor. Esta *gateway* não é mais do que um *Bus* de informação que recebe informações de dispositivos que estão dentro de uma determinada habitação e publica essa informação no “AALNode”.

No momento da realização deste cenário de teste o projeto AAL4ALL ainda não possuía especificações normalizadas para a criação de *gateways*, existindo diferentes *gateways* desenvolvidas por vários parceiros que comunicam utilizando diferentes interfaces. Por essa razão, e uma vez que o processo de certificação das *gateways* terá de ser realizado no futuro sobre uma *gateway* normalizada e ainda devido à falta de documentação que especificasse de forma clara as comunicações existentes entre o *driver* e a própria *gateway*, o conjunto *Gateway + Driver + Sensor* foi aqui utilizado como um componente único.

4.2 Análise da Documentação dos Produtos e Serviços

Como já referido no capítulo 3, o projeto AAL4AAL já possuía um *template* de documentação de componentes à data da realização deste cenário de teste, o qual foi utilizado pelos fabricantes para descreverem os seus componentes. Estes documentos estão alojados no repositório *skydrive* do projeto AAL4AAL, sendo por isso o primeiro passo a recolha dos documentos relativos aos componentes que integram o cenário de teste. Para esse efeito foram recolhidos os documentos de descrição do “AALNode”, “Caretaker Application” e “INESC AALGateway”. Como passo seguinte, foram analisados cada um destes documentos, no sentido de compreender os seus casos de

uso e as suas interfaces, sendo estes dois elementos fundamentais para a especificação dos testes a realizar sobre cada um dos componentes. Realizada essa primeira análise, constatou-se que em todos os documentos havia omissões de informações, ou porque não descreviam de forma correta os casos de uso do componente em questão, ou porque simplesmente não possuíam informações concretas sobre o conteúdo das mensagens que eram trocadas através das suas interfaces. Esta situação devia-se, em alguns casos, ao facto de alguns dos produtos ainda estarem descritos segundo o documento antigo e não sobre o *template* mais recente (ver Secção 3.2).

Uma vez identificados estes problemas, foi solicitado aos fabricantes dos componentes que documentassem novamente os seus produtos. Ao longo desta fase, foi ainda necessário interagir com os fabricantes, no sentido de os ajudar a documentar de forma detalhada os seus produtos uma vez que alguns dos fabricantes tinham dúvidas sobre algumas secções do *template*. Os documentos resultantes deste processo foram então utilizados para especificar os teste a realizar sobre cada um dos componentes.

4.3 Redocumentação

Apesar da interação realizada com os fabricantes dos componentes a integrar no cenário de teste, a documentação final dos documentos ainda apresentava omissões e ambiguidades, pelo que foi necessário documentar com mais formalismo os vários componentes. O primeiro componente alvo desta redocumentação foi o “AALNode”, pois este componente apresentava omissões nomeadamente ao nível das suas interfaces. A documentação mais formal das interfaces do componente “AALNode” pode ser observada na Tabela 4.2.

Tabela 4.1: Requisitos identificados para o “AALNode”

Requirement Type	Description	Identifier
Functional Requirements	The application should allow the subscription of messages (belonging to a particular set of topics).	AFR.1
	The application should allow the publication of messages, and forward them to subscribers.	AFR.2
Non-functional requirement - Performance efficiency	The application should forward messages with a time delay less than 200 ms.	ANFR.1
Non-functional requirement - Security	The application should use authentication mechanisms for subscription and publication of messages.	ANFR.2
Non-functional requirement - Reliability	The application should ensure delivery of messages, even when a subscriber is offline.	ANFR.3

Após a documentação das interfaces do componente “AALNode”, foi necessário documentar de forma mais formal as interfaces dos restantes componentes do cenário piloto.

Além das interfaces, outras partes dos documentos mereceram também uma redocumentação mais formal, nomeadamente a identificação de requisitos a partir da descrição dos casos de uso onde foi, por exemplo, necessário incluir identificadores. A Tabela 4.1 apresenta os requisitos

Tabela 4.2: Descrição mais formal do componente “AALNode”

Name	AALNode
Description	Provide system integration capabilities, acquiring data from components and forwarding it to the proper destinations, according to internal rules and validations.
Kind	Software Application
Interfaces	name: PublisherInterface description: Interface that receives publication requests. kind: SoftwareInterface
	name: SubscriberInterface description: Interface that receives subscription requests. kind: SoftwareInterface.
	name: CallbackInterface description: Interface to be implemented by subscribers, so that the AALNode can forward published messages. kind: SoftwareInterface.
State variables	Table of subscriptions. It is a set of pairs (Topic, Subscriber).
Receives	message: Subscribe(int ID, list<string>ListOfTopics) interface: SubscriberInterface enable condition: always enabled effect: Update table of subscriptions
	message: Publish(string Origin, string Topic, string Payload) interface: PublisherInterface enable condition: always enabled effect: send Publish(Origin, Topic, Payload) for all subscribers in table of subscriptions with the same topic
Sends	message: Publish(string Origin, string Topic, string Payload) interface: CallbackInterface trigger: receive Publish(string Origin, string Topic, string Payload)
Realization options	Ethernet, AMQP (for interface SubscriberInterface and CallbackInterface) and HTTP (PublisherInterface)
Prerequisites	-
Non-funct. requirements	Message sending delay is less than 200 ms. (ANFR.1)

identificados para o componente “AALNode”. Os requisitos não funcionais foram classificados de acordo com a norma ISO/IEC 25010:2011 (ver secção 2.1.2).

4.4 Especificação dos Testes

Analisando a documentação de cada um dos componentes que integram o cenário de teste, nomeadamente os seus casos de uso e seguindo o processo de teste e certificação apresentado no capítulo 3, foram inicialmente especificados os testes de conformidade para cada um dos componentes. Essa especificação foi realizada segundo a norma IEEE Std 829TM-2008 [IEE08], tendo sido criado para isso um ficheiro Excel no repositório skydrive do projeto AAL4AAL onde foram criadas diferentes folhas para cada um dos componentes.

O primeiro componente para o qual foi elaborada a especificação foi o “AALNode”, tendo essa especificação sido realizada a partir dos requisitos identificados para este componente (Tabela 4.1). Os casos de teste especificados para o componente “AALNode”, os quais estão representados na Tabela 4.3, foram elaborados de forma a cobrir, tanto as situações normais de envio/receção de mensagens como também as situações limite onde normalmente são identificados os principais problemas, sendo que um determinado caso de teste pode corresponder ao teste de um ou mais requisitos.

A especificação dos casos de teste, baseada nos requisitos identificados foi igualmente realizada para os restantes componentes do cenário de teste estando essas especificações representadas na Tabela 4.4 e Tabela 4.5, para a aplicação “Caretaker Application”, e na Tabela 4.6 e Tabela 4.7, para o conjunto *Gateway + Driver + Sensor*.

Tal como realizado para os componentes, também foi necessário especificar os casos de teste sobre o cenário de integração onde participaram todos os componentes. Para isso, foi necessário identificar os requisitos, desta vez sobre o cenário de integração. Neste caso, o cenário diz respeito a um serviço de deteção de atividade, tendo sido identificados os requisitos constantes da Tabela 4.8. A partir destes requisitos foram especificados os casos de teste descritos na Tabela 4.9. O objetivo destes testes é assegurar que os componentes funcionam corretamente em conjunto.

Apesar de nesta experiência os requisitos terem sido identificados a partir das especificações dos componentes, é esperado que após terem sido criadas diferentes classes de componentes, sejam elaborados pela Associação AAL4ALL um conjunto de requisitos e especificações para cada uma dessas classes, sendo posteriormente elaborada pela Entidade Certificadora a especificação dos testes necessários para cada componente de forma a garantir que o mesmo cumpre os pré-requisitos de cada uma das classes para o qual se candidata a obter certificação.

Tabela 4.3: Especificação dos testes de conformidade para o “AALNode”

Test case identifier	Objective	Inputs	Outcome(s)	Requirement identifier
Nod.T1	Check if a message (belonging to a particular topic) is received by a subscriber.	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) with: msg1= '{ "data": "text XXXXXXXX" }' topic1= 'health.devices.1'	Subscriber 1 receive msg1	AFR.1, AFR.2
Nod.T2	Check if multiple published messages (belonging to a particular topic) are received by their subscriber	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) Publish(3, topic1 , msg2) Publish(4, topic1 , msg3) with: topic1= 'health.devices.1' msg1= '{ "data": "text XXXXXXXX" }' msg2= '{ "data": "text XXXXXXXX" }' msg3= '{ "data": "text XXXXXXXX" }'	Subscriber 1 receive msg1, msg2 and msg3	AFR.1, AFR.2
Nod.T3	Check if a published message (belonging to a particular topic) is received by multiple subscribers.	Subscribe(1, [topic1]) Subscribe(2, [topic1]) Subscribe(3, [topic1]) Publish(4, topic1 , msg1) with: msg1= '{ "data": "text XXXXXXXX" }' topic1= 'health.devices.1'	Subscribers 1,2,3 receive msg1	AFR.1, AFR.2
Nod.T4	Check if a published message is sent when there's no subscriber	Publish(2, topic1 , msg1) with: msg1= '{ "data": "text XXXXXXXX" }' topic1= 'health.devices.1'	Discard msg1	AFR.1, AFR.2
Nod.T5	Check if an old message is not sent or forwarded	Publish(2, topic1 , msg1) Subscribe(1, [topic1]) Publish(2, topic1 , msg2) with: topic1= 'health.devices.1' msg1= '{ "data": "t+1" }' msg2= '{ "data": "t-1" }'	Subscriber 1 receive msg2	AFR.1, AFR.2
Nod.T6	Check if two identical messages (belonging to the same topic) are both received	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) Publish(2, topic1 , msg1) with: msg1= '{ "data": "text XXXXXXXX" }' topic1= 'health.devices.1'	Subscriber 1 receive msg1 (in duplicate)	AFR.1, AFR.2
Nod.T7	Check if two messages (belonging to the same topic) are received in the same delivering order	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) Publish(2, topic1 , msg2) Publish(2, topic1 , msg3) with: topic1= 'health.devices.1' msg1= '{ "data": "text XXXXXXXX" }' msg2= '{ "data": "text XXXXXXXX" }' msg3= '{ "data": "text XXXXXXXX" }'	Subscriber 1 receives msg1 msg 2 msg 3	AFR.1, AFR.2
Nod.T8	Check if empty messages are sent or received	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) with: msg1= '{ "" }' topic1= 'health.devices.1'	Discard msg1	AFR.1, AFR.2
Nod.T9	Check how long it takes to receive a message (response time)	-	-	ANFR.1
Nod.T10	Check how long takes to send a message (response time)	-	-	ANFR.1
Nod.T11	Check if any message is lost or received in asynchronous mode when the system is down	Subscribe(1, [topic1]) Publish(2, topic1 , msg1) with: msg1= '{ "data": "text XXXXXXXX" }' topic1= 'health.devices.1'	Subscriber 1 receives msg1	ANFR.3
Nod.T12	Check if there is an authentication and authorization control mechanism for subscribers	-	-	ANFR.2

Tabela 4.4: Requisitos identificados para a aplicação “Caretaker Application”

Requirement Type	Description	Identifier
Functional Requirements	The application must allow the reception of data from the AALNode.	CAFR.1
	The application must show to the user the information received from AALNode.	CAFR.2
Non-functional requirement - Security	The application must ensure that Caretakers only receive information of your users.	CANFR.1

Tabela 4.5: Especificação dos testes de conformidade para a aplicação “Caretaker Application”

Test case identifier	Objective	Inputs	Outcome(s)	Requirement identifier
CA.T1	Check if a message (sent by AALNode) is shown to the end user.	Publish(id1, topic1, msg1) with: id1= 1 msg1= 'payload': 'type': 1, 'state': 0, 'sensor': 1 topic1= 'aj.FhP.MultiSensor' Type 1 is a Door and State 0 is Off	A message to the user through a GUI, that can be abstractly described according to the signature displaySensorState(time, SensorType, SensorId, State) with: time= current system time SensorType= Door SensorId= 1 State= Off	CAFR.1, CAFR.2
CA.T2	Check if multiple messages (sent by AALNode) corresponding to the same type of sensor but with different states are presented to the end user.	Publish(id1, topic1, msg1) Publish(id1, topic1, msg2) Publish(id1, topic1, msg3) with: id1= 1 topic1= 'aj.FhP.MultiSensor' msg1= 'payload': 'type': 1, 'state': 1, 'sensor': 1 msg2= 'payload': 'type': 1, 'state': 0, 'sensor': 1 msg3= 'payload': 'type': 1, 'state': 1, 'sensor': 1	displaySensorState(time, SensorType1, SensorId1, State1) displaySensorState(time, SensorType1, SensorId1, State2) displaySensorState(time, SensorType1, SensorId1, State1) with: time= current system time SensorType1= Door SensorId1= 1 State1= On State2= Off	CAFR.1, CAFR.2
CA.T3	Check if multiple messages (sent by AALNode) corresponding to the different types of sensors are presented to the end user.	Publish(id1, topic1, msg1) Publish(id1, topic1, msg2) Publish(id1, topic1, msg3) with: id1= 1 topic1= 'aj.FhP.MultiSensor' msg1= 'payload': 'type': 0, 'state': 0, 'sensor': 1 msg2= 'payload': 'type': 1, 'state': 0, 'sensor': 1 msg3= 'payload': 'type': 2, 'state': 0, 'sensor': 1 Type 0 is Bed Pressure and type 2 is Window	displaySensorState(time, SensorType1, SensorId1, State1) displaySensorState(time, SensorType2, SensorId1, State1) displaySensorState(time, SensorType3, SensorId1, State1) with: time= current system time SensorType1= Bed Pressure SensorType2= Door SensorType3= Window SensorId1= 1 State1= Off	CAFR.1, CAFR.2
CA.T4	Check if there is privacy control of the data received.	Publish(id1, topic1, msg1) with: id1= 1 topic1= 'aj.FhP.MultiSensor' msg1= 'payload': 'type': 1, 'state': 0, 'sensor': 1 from user A	The caretaker B, who have no access to information of User A should not view the information	CANFR.1

Tabela 4.6: Requisitos identificados para o conjunto Gateway + Driver + Sensor

Requirement Type	Description	Identifier
Functional Requirements	Should detect changes in the status of doors, windows and beds.	CGDSFR.1
	Must send the data to the AALNode.	CGDSFR.2

Tabela 4.7: Especificação dos testes de interoperabilidade para o conjunto Gateway + Driver + Sensor

Test case identifier	Objective	Inputs	Outcome(s)	Requirement identifier
CGDS.T1	Check if a sensor signal is received in AALNode.	Change the bed sensor state	AALNode received msg1. msg1 = 'Senior1.FhP.MultiSensor' : ''sensor':1,'state':0,'type':0'	CGDSFR.1, CGDSFR.2
CGDS.T2	Check if a sensor signal is received in AALNode.	Change the door sensor state	AALNode received msg1. msg1 = 'Senior1.FhP.MultiSensor' : ''sensor':1,'state':1,'type':1'	CGDSFR.1, CGDSFR.2
CGDS.T3	Check if a sensor signal is received in AALNode.	Change the window sensor state	AALNode received msg1. msg1 = 'Senior1.FhP.MultiSensor' : ''sensor':1,'state':0,'type':2'	CGDSFR.1, CGDSFR.2

Tabela 4.8: Requisitos identificados para o cenário de teste

Requirement Type	Description	Identifier
Functional Requirements	The service must be able to detect activity.	CFR.1
	The service should show the user's movement to the caretaker through the application.	CFR.2
Non-functional requirement - Security	The service should show the user's movement only to the caretaker responsible for his residence.	CNFR.1

Tabela 4.9: Especificação dos testes de interoperabilidade para o cenário de teste

Test case identifier	Objective	Inputs	Outcome(s)	Requirement identifier
InSc1.T1	Check if a sensor signal appears in the application user interface.	Close the door	A message with the time, the sensor type and state is displayed to the end user.	CFR.1, CFR.2
InSc1.T2	Check if the changes of the sensor state appear in the application user interface.	Close the door, open the door and close the door again	The 3 messages with the time, the sensor type and state are displayed to the end user	CFR.1, CFR.2
InSc1.T3	Check if a sensor signal appears in the application user interface.	Close the window	A message with the time, the sensor type and state is displayed to the end user.	CFR.1, CFR.2
InSc1.T4	Check if the changes of the sensor state appear in the application user interface.	Close the window, open the window and close the window again	The 3 messages with the time, the sensor type and state are displayed to the end user	CFR.1, CFR.2
InSc1.T5	Check if a sensor signal appears in the application user interface.	Person rises from the bed	A message with the time, the sensor type and state is displayed to the end user.	CFR.1, CFR.2
InSc1.T6	Check if the changes of the sensor state appear in the application user interface.	Person lies down on the bed and after two minutes gets up from bed	The 2 messages with the time, the sensor type and state are displayed to the end user	CFR.1, CFR.2
InSc1.T7	Check if there is privacy control of the data sent.	Close the door	The caretaker B, who no access to information of User A should not view the information of the change of status of the door.	CNFR.1

```

1 class Sender:
2
3     def __init__(self, idd2, topic, payload, FDG):
4
5         target = 'http://bruxelix.inescporto.pt:8080/publish/'
6         data = '{ "origin": "user.5423225.device.2432", "date": "232425326235", "topic
           ": ' + topic + ', "payload": ' + payload + ' }'
7         r = requests.post(
8             target,
9             data=data,
10            auth=HTTPBasicAuth('test', 'test'))

```

Listagem 4.1: Class Sender

4.5 Implementação dos *Scripts* de Testes

De forma a automatizar o processo de teste, e como descrito no capítulo 3, sempre que possível deve-se criar plataformas de teste que permitam, não só a automatização do processo, como o reemprego dessas mesmas plataformas para a realização dos testes sobre outros componentes. Neste caso, e tomando novamente como exemplo o componente “AALNode”, foi elaborado um conjunto de *scripts* em Python [Fou13] que permitem dessa forma a automatização dos testes. Por forma a facilitar a criação simplificada dos *scripts* de teste foram criados ficheiros que fornecem as classes e funções necessárias a todos os testes e também à forma de como avaliar o seu resultado. As principais classes são:

- A classe *Subscriber* que se encontra descrita na Listagem 4.2 que estabelece a ligação com o “AALNode” ficando posteriormente à escuta, esperando por mensagens relativas aos tópicos que subscreveu;
- A classe *Sender*, descrita na Listagem 4.1 e que é responsável pelo envio de mensagens para o “AALNode”;
- Conjunto de funções que se encontram detalhadas na Listagem 4.3, que ajudam na criação dos ficheiros necessário à avaliação do resultado dos testes.

Uma vez criadas estas classes e funções a criação de um *script* é um processo simples e de quase transposição direta entre a especificação e a sua implementação. Um exemplo disso mesmo pode ser observado na Listagem 4.4 que contém a descrição do *script* do teste Nod.T7 especificado na Tabela 4.3 da secção anterior.

A forma de avaliação de resultados de teste utilizada foi a comparação de ficheiros de texto, sendo que para cada subscritor é criado à partida um ficheiro com as mensagens que é esperado ele receber durante o teste. Durante a execução do teste as mensagens recebidas por cada subscritor são guardadas no seu respetivo ficheiro de texto sendo posteriormente comparado o conteúdo desse ficheiro com o ficheiro que contém o resultado esperado. Após a criação de todos os *scripts*

```

1 class Subscriber:
2
3     def __init__(self, idd2, binding_keys, TextName, FDG):
4
5         self.idd = idd2
6         connection = pika.BlockingConnection(
7             pika.ConnectionParameters(
8                 host='bruxelix.inescporto.pt')
9             )
10        channel = connection.channel()
11        channel.exchange_declare(
12            exchange='aal4all',
13            type='topic')
14        result = channel.queue_declare(exclusive=True, durable=True)
15        queue_name = result.method.queue
16        # Subscribe each binding key in the exchange
17        for binding_key in binding_keys:
18            channel.queue_bind(
19                exchange='aal4all',
20                queue=queue_name,
21                routing_key=binding_key)
22        name = "tmp/" + TextName + "_Subscriber" + str(idd2) + ".txt"
23        if (FDG == 1):
24            print 'Subscriber ' + str(idd2) + ' Waiting for data. To exit press
                CTRL+C'
25        def callback(ch, method, properties, body):
26            if (FDG == 1):
27                print " [x] %r:%r" % (method.routing_key, body,) + " Receiver for
                    Subscriber " + str(idd2)
28            try:
29                f = open(name, "a")
30                f.write("%r:%r\n" % (method.routing_key, body,))
31                f.close()
32            except:
33                pass
34        channel.basic_consume(
35            callback,
36            queue=queue_name,
37            no_ack=True)
38        thread.start_new_thread(channel.start_consuming, ())

```

Listagem 4.2: Class Subscriber

```

1  # Create the check file line
2  def creatfileline (name, y, z):
3      Text = textforamenssagepayload(z, y)
4      top = topicname = "'health.devices.'" + str(z) + "'"
5      try:
6          f = open(name, "a")
7          f.write("%s:%s\n" % (top, Text))
8          f.close()
9      except:
10         pass
11
12 def CreatExpFile (TestName, SN, ListText):
13
14     name = "ExpFiles/" + TestName + "_Subscriber" + str(SN) + "_EXP.txt"
15     for x in range(0, len(ListText)):
16         temp = ListText[x]
17         top = "'" + temp [0] + "'"
18         Text = "'" + temp[1] + "'"
19         try:
20             f = open(name, "a")
21             f.write("%s:%s\n" % (top, Text))
22             f.close()
23         except:
24             pass
25
26 # Delete all files for one subscriber
27 def deleteallfiles(sn, TestName):
28     for x in range(1, (sn + 1)):
29         namef1 = "ExpFiles/" + TestName + "_Subscriber" + str(sn) + "_EXP.txt"
30         namef2 = "tmp/" + TestName + "_Subscriber" + str(sn) + ".txt"
31         if os.path.isfile(namef1):
32             os.remove(namef1)
33         if os.path.isfile(namef2):
34             os.remove(namef2)
35
36 # check the result using file comparison
37 def checkresult(sn, TestName):
38
39     for x in range(1, (sn + 1)):
40
41         tr = 0
42         namef1 = "ExpFiles/" + TestName + "_Subscriber" + str(sn) + "_EXP.txt"
43         namef2 = "tmp/" + TestName + "_Subscriber" + str(sn) + ".txt"
44
45         if filecmp.cmp(namef1, namef2):
46             #if (FDG == 1):
47             #print "All messages for Subscriber " + str(sn) + " were successfully received
48             !"
49             deleteallfiles(x, TestName)
50         else:
51             #if (FDG == 1):
52             #print "Subscriber " + str(sn) + " has not received all menssages!"
53             tr = 1
54             #print str(tr)
55     return tr

```

Listagem 4.3: Conjunto de funções auxiliares

```

-----
AALNode Test - M E N U
-----
1. Run All Tests
2. Run a particular test
-----
Enter your choice [1-2] : 1
Starting run all tests...
Starting run NodT1...
NodT1 PASSED! All messages have been successfully received by their subscribers
Starting run NodT2...
NodT2 PASSED! All messages have been successfully received by their subscribers
Starting run NodT3...
NodT3 PASSED! All messages have been successfully received by their subscribers
Starting run NodT4...
NodT4 PASSED! The message was correctly rejected!
Starting run NodT5...
NodT5 PASSED! All messages have been successfully received by their subscribers
Starting run NodT6...
NodT6 PASSED! All messages have been successfully received by their subscribers
Starting run NodT7...
NodT7 PASSED! All messages have been successfully received (In the correct order
) by their subscribers

```

Figura 4.2: Representação da interface do script de execução de teste do “AALNode”

correspondentes aos testes especificados, foi ainda elaborado um *script* que permite, mesmo a utilizadores sem conhecimentos de programação executarem não só um teste particular mas também todo o conjunto de testes, de forma intuitiva. A Figura 4.2 apresenta a interface desse *script*.

Como mais nenhum dos componentes integrados neste cenário de teste possui apenas interfaces de software, não foi possível automatizar completamente mais nenhum processo de teste. No entanto os *scripts* criados para o componente “AALNode”, foram reutilizados para enviar mensagens para a aplicação “Caretaker Application” e para confirmar o envio de mensagens por parte do conjunto *Gateway + Driver + Sensor*. A avaliação dos resultados dos testes foi, neste caso efetuada através da verificação de evidências, como as representadas na Figura 4.3 e Figura 4.4, onde constam a apresentação de uma mensagem enviada pelo conjunto *Gateway + Driver + Sensor* e recebida pelo “AALNode” e a apresentação na aplicação “Caretaker Application” da recepção de dados provenientes do “AALNode” respectivamente. Com a utilização destas duas formas de teste (totalmente e parcialmente automatizada), foi possível obter uma maior eficiência, do que se fossem utilizados testes manuais em todos os casos, uma vez que desta forma é simples não só executar várias vezes um conjunto de testes, como também é mais rápido todo esse processo.

```

→ AALNODE_TEST python consumir.py
[*] Waiting for data. To exit press CTRL+C
[x] 'Senior1.FhP.MultiSensor': '{"sensor":1,"state":0,"type":2}'

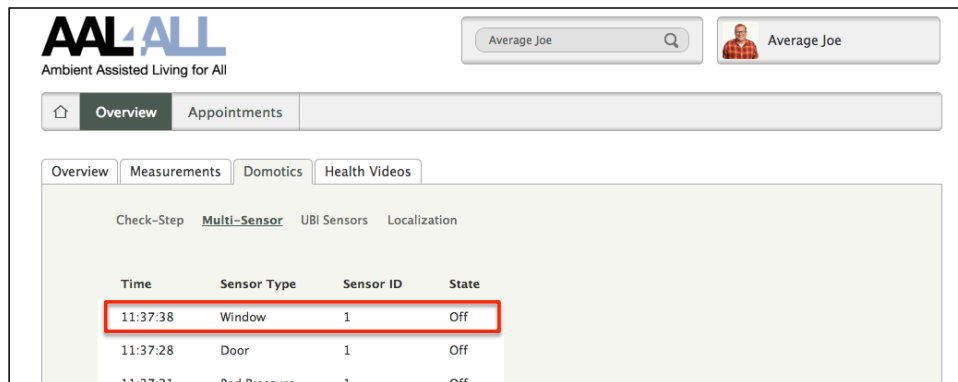
```

Figura 4.3: Representação da receção de informação por parte do “AALNode”

Aplicação e Validação da Metodologia

```
1  TestName = 'NodT7'
2  print "Starting run " + TestName + "..."
3
4  # Create Expected File
5  CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 1"}'], ['health.devices.1', '{"data":"binary payload 2"}'], ['health.devices.1', '{"data":"binary payload 3"}']])
6
7  # Activate Subscribers
8  Subscriber(1, ['health.devices.1'], TestName, FDG)
9
10 # Send Menssages
11 Sender(1, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
12 Sender(2, 'health.devices.1', '{"data":"binary payload 2"}', FDG)
13 Sender(3, 'health.devices.1', '{"data":"binary payload 3"}', FDG)
14
15 # Check Result
16 # print the test result
17 if (checkresult(1, TestName) == 0):
18     print TestName + " PASSED! All messages have been successfully received (In the correct order) by their subscribers"
19 else:
20     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in temp and ExpFiles folders for more informations"
```

Listagem 4.4: Script do teste Nod.T7



The screenshot shows the AAL4ALL web application interface. At the top, there is a header with the AAL4ALL logo and the text "Ambient Assisted Living for All". To the right of the header, there is a search bar with the text "Average Joe" and a user profile icon. Below the header, there is a navigation bar with tabs for "Overview", "Measurements", "Domotics", and "Health Videos". The "Overview" tab is selected. Below the navigation bar, there is a section titled "Check-Step" with sub-tabs for "Multi-Sensor", "UBI Sensors", and "Localization". The "Multi-Sensor" sub-tab is selected. Below the sub-tabs, there is a table with the following columns: "Time", "Sensor Type", "Sensor ID", and "State". The table contains three rows of data:

Time	Sensor Type	Sensor ID	State
11:37:38	Window	1	Off
11:37:28	Door	1	Off
11:37:21	Bed Pressure	1	Off

Figura 4.4: Representação da recepção de informação por parte da aplicação “Caretaker Application”

4.6 Resultados da Execução

Os primeiros testes a serem executados foram os testes sobre o componente “AALNode”, uma vez que este é o componente central deste cenário de teste e porque uma vez assegurado o seu correto funcionamento poderá ser utilizado de forma a facilitar os testes dos restantes componentes. Os resultados obtidos dos testes realizados sobre o componente “AALNode” encontram-se descritos na Tabela 4.10. Olhando para essa mesma tabela é facilmente perceptível que foram detetados

alguns problemas neste componente, uma vez que só os primeiros sete testes obtiveram resultado *Passed*. O teste Nod.T8, obteve o resultado *Inconclusive*, uma vez que apesar de o resultado do teste não estar de acordo com o resultado esperado constante da especificação, não é claro se o “AALNode” deve ou não descartar uma mensagem vazia. Uma vez que este componente não analisa o conteúdo das mensagens, pode ser permitido o reenvio de mensagens vazias. Os testes Nod.T9 e Nod.T10, obtiveram ambos o resultado *Not tested*, porque tanto na documentação do componente como na especificação dos testes não está descrito em que condições de ambiente é que o componente deve garantir a entrega de mensagens dentro de um determinado tempo. Por último, os testes Nod.T11 e Nod.T12, obtiveram o resultado *Failed*, uma vez que no primeiro caso não é garantida a receção de mensagens no caso de um subscritor não estar *online* e no segundo caso porque não existe nenhum mecanismo de segurança que assegure que só entidades com determinadas credenciais podem receber mensagens de determinados tópicos.

Tabela 4.10: Resultado da execução dos testes sobre o “AALNode”

Test case identifier	Result	Notes
Nod.T1	Passed	-
Nod.T2	Passed	-
Nod.T3	Passed	-
Nod.T4	Passed	-
Nod.T5	Passed	-
Nod.T6	Passed	-
Nod.T7	Passed	-
Nod.T8	Inconclusive	The message isn't discarded. The receiver, received the empty message. Check what is the intended behavior.
Nod.T9	Not tested	Need to clarify load profile and operational environment.
Nod.T10	Not tested	Need to clarify load profile and operational environment.
Nod.T11	Failed	Receivers don't have persistent ids, if it goes offline and then online, the messages will be lost.
Nod.T12	Failed	There is no such mechanism. This was verified by analysing the interfaces.

Embora os resultados obtidos pelo componente “AALNode”, num contexto de aplicação real da metodologia de teste e certificação proposta fizessem com que o componente fosse desde já excluído do processo de certificação, foi decidido, no contexto desta experiência e tendo em conta que os resultados *Not tested* e *Failed* foram apenas obtidos em testes a requisitos não funcionais (que não colocam em causa o funcionamento do componente), continuar a experiência mantendo este componente e assumindo que ele passou a etapa dos testes de conformidade. Assim sendo, o processo de teste e certificação continua com a execução de teste de conformidade, agora sobre a aplicação, “Caretaker Application”.

Tabela 4.11: Resultado da execução dos testes sobre a aplicação “Caretaker Application”

Test case identifier	Result	Notes
CA.T1	Passed	Failed in first test execution, subsequently fixed.
CA.T2	Passed	Failed in first test execution, subsequently fixed.
CA.T3	Passed	Failed in first test execution, subsequently fixed.
CA.T4	Failed	Although the application has a mechanism to control users, all Caretakers receive information of all users.

Neste caso (como demonstra a Tabela 4.11), os três primeiros testes obtiveram o resultado *Passed*, mas apenas numa segunda execução e após um diálogo com o fabricante da aplicação no sentido de este corrigir erros que foram detetados na primeira execução dos testes, onde todas as mensagens enviadas para a aplicação eram apresentadas em duplicado e com horário diferente aos utilizadores. O quarto teste obteve o resultado *Failed*, uma vez que apesar de a aplicação possuir mecanismo de controlo de utilizadores, esta não permite na sua versão atual uma filtragem das informações recebidas, o que faz com que todos os utilizadores possam visualizar todas as informações que são recebidas pela aplicação. Obviamente que mais uma vez e tal como explicado no caso do “AALNode”, este resultado implicaria a exclusão desta aplicação do processo de teste e certificação, mas pela mesma razão, foi decidido continuar a experiência, assumindo que a aplicação completou com sucesso a fase dos testes de conformidade.

Por último foram realizados testes sobre o conjunto *Gateway + Driver + Sensor*, estando os resultados desses testes representados na Tabela 4.12.

Tabela 4.12: Resultado da execução dos testes sobre o conjunto Gateway + Driver + Sensor

Test case identifier	Result	Notes
CGDS.T1	Passed	-
CGDS.T2	Passed	-
CGDS.T3	Passed	-

Como se pode constatar todos os testes obtiveram o resultado *Passed*, o que indica que este conjunto também concluiu com sucesso a fase dos testes de conformidade. Uma vez concluída a fase dos testes de conformidade sobre todos os produtos que integram este cenário de testes, foram realizados os testes de interoperabilidade, cuja especificação está contida na Tabela 4.9. Os resultados, representados na Tabela 4.13, mostram que os seis primeiros testes realizados sobre o cenário piloto obtiveram o resultado *Passed*, no entanto o sétimo teste obteve o resultado *Failed*, uma vez que independentemente do utilizador simulado, as suas informações eram apresentadas ao *Caretaker*.

Tabela 4.13: Resultado da execução dos testes sobre o cenário

Test case identifier	Result	Notes
InSc1.T1	Passed	-
InSc1.T2	Passed	-
InSc1.T3	Passed	-
InSc1.T4	Passed	-
InSc1.T5	Passed	-
InSc1.T6	Passed	-
InSc1.T7	Failed	All Caretakers receive information of all users

4.7 Resultados do Processo de Certificação

Com base nos resultados dos testes de conformidade e interoperabilidade apresentados na subsecção anterior, não é possível emitir qualquer certificado AAL4ALL para os componentes testados, pois todos os componentes revelaram falhas no processo de teste. Os componentes “AAL-Node” e a aplicação “Caretaker Application”, falharam inclusive na fase dos testes de interoperabilidade, ou seja para cada um destes componentes, foi emitido um relatório de não conformidade, onde são reportados ao fabricante os testes em que o seu componente não passou. O relatório de não conformidade do componente “AALNode” encontra-se representado na Figura 4.5.

Quanto ao conjunto Gateway + Driver + Sensor, apesar de ter realizado com sucesso os testes de conformidade, não é possível garantir que este componente funcione corretamente num cenário de integração, uma vez que os restantes componentes integrantes desse cenário apresentaram falhas que inclusive fizeram com que o último teste sobre o cenário obtivesse o resultado de *Failed*. Desta forma não é garantido que estes componentes sejam capazes de comunicar entre si de forma a garantir corretamente o serviço simulado neste cenário piloto, não estando desta forma aptos a serem colocados no mercado, onde iriam comunicar com outros componentes do ecossistema AAL4ALL.

A aplicação da metodologia no cenário de teste descrito permitiu validar e refinar a mesma. Por outro lado, quer a metodologia proposta quer a forma de aplicação foram alvo de validação junto dos restantes parceiros do projeto AAL4ALL, primeiro a nível interno, em reuniões do PPS5 realizadas nas instalações da Faculdade de Engenharia da Universidade do Porto e nas instalações do CITEVE - Centro Tecnológico das Indústrias Têxtil e do Vestuário de Portugal a 25 de Setembro de 2013 e a 26 de Novembro de 2013 respetivamente. E posteriormente validadas na reunião geral do consórcio AAL4ALL realizada nas instalações do IPN - Instituto Pedro Nunes, em Coimbra, no dia 29 de Novembro de 2013.

Non-conformance Report

Product Identification:

Manufacturer: INESC Porto

Product: AALNode

Version: 1.0

Test Laboratory

Organization x

Test Date

24/11/2013

Step of the testing and certification process:

Conformance tests

Problems:

Test case identifier	Objective	Inputs	Outcome(s)	Result	Notes
Nod.T8	Check if empty messages are sent or received	Subscriber 1, {topic 1}: Send2, topic 1, msg 1} with msg = "1", topic = "health.devices 1"	discard msg 1	Inconclusive	The message isn't discarded. The receiver, received the empty message. Check what is the intended behavior.
Nod.T9	Check how long takes to receive a message (response time)			Not tested	Need to clarify load and operational environment.
Nod.T10	Check how long takes to send a message (response time)			Not tested	Need to clarify load and operational environment.
Nod.T11	Check if any message is lost or received in asynchronous mode when the system is down	Subscriber 1, {topic 1}: Send2, topic 1, msg 1} with msg = "1", "data","test XXXXXXXX", topic = "health.devices 1"	Subscriber 1 receive msg 1	Failed	Receivers don't have persistent ids, if it goes offline and then online, the messages will be lost.
Nod.T12	Check if there is an authentication and authorization control mechanism for subscribers			Failed	There is no such mechanism. This was verified by analysing the interfaces.

Observation:

The manufacturer must fix the problems and answer questions presented.

Figura 4.5: Relatório de não conformidade emitido para o “AALNode”

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as conclusões do presente projeto de dissertação, nomeadamente a análise à satisfação dos objetivos inicialmente propostos. São ainda apresentadas algumas ideias, sobre o que poderá ser o trabalho futuro quer em termos do projeto AAL4ALL ou mesmo um trabalho mais académico.

5.1 Satisfação dos Objetivos

Todos os objetivos inicialmente propostos foram alcançados no decorrer deste projeto de dissertação. Foi elaborada uma metodologia de teste e certificação para um ecossistema de *Ambient Assisted Living*, baseada no modelo de negócio do projeto AAL4AAL e que consiste em três principais fases, análise de pré-requisitos, testes de conformidade e testes de interoperabilidade. É no entanto importante salientar que devido ao facto de este projeto de dissertação estar inserido num projeto nacional, como é o caso do projeto AAL4AAL, e ao facto do mesmo envolver muitos parceiros, nem sempre foi fácil conseguir ter em tempo útil, quer as especificações dos componentes ou mesmo os próprios componentes para teste. Contudo e apesar destas dificuldades a metodologia foi validada através da sua aplicação num cenário de teste piloto utilizando produtos reais do projecto AAL4AAL, tendo ainda sido criada uma plataforma automática de testes que permitiu automatizar e optimizar os testes realizados sobre os componentes.

Ao longo do teste piloto foi necessário tomar algumas decisões, nomeadamente permitir que um componente que falhou os testes de conformidade continuasse para a fase dos testes de interoperabilidade, pois só assim seria possível chegar à fase dos testes de interoperabilidade. Esta avaliação da metodologia, permitiu identificar problemas não só na documentação dos componentes como também problemas reais dos próprios componentes, revelando-se desse modo útil não só na garantia de interoperabilidade, mas também na ajuda à identificação de erros dos próprios componentes.

As opções tomadas ao longo da definição desta metodologia e a sua aplicação num cenário de teste têm impacto significativo num conjunto de atividades e parceiros do projeto AAL4ALL e, espera-se, no pós-projeto, que a metodologia proposta seja a base de um ecossistema que continuará para além do projeto AAL4ALL.

5.2 Trabalho Futuro

Como trabalho futuro, obviamente dependente das decisões do consórcio AAL4ALL, tal como já anteriormente referido no capítulo 3 será necessário criar categorias de componentes, de forma a que possam ser elaboradas listas de pré-requisitos para os componentes candidatos. De igual modo após a criação dessas mesmas categorias, será possível definir um conjunto de requisitos, que os componentes candidatos terão de cumprir, de forma a que a elaboração dos testes de conformidade a serem realizados sobre esses componentes sejam criados tendo como objetivo garantir que os componentes candidatos cumprem os requisitos da(s) categoria(s) à(s) qual(uais) se candidata(m). Essas categorias de componentes, devem também possuir um conjunto de serviços, serviços esses que possam ser transformados em cenários de teste a serem utilizados nos testes de interoperabilidade, tornando assim estes testes mais direcionados às funções reais que os componentes vão desempenhar quando ingressarem no ecossistema. Será também interessante tentar conseguir uma normalização das interfaces ao nível das categorias de componentes, uma vez que isso permite a criação de estruturas automáticas de testes mais fáceis de manter e que até podem ser cedidas aos próprios fabricantes de produtos de modo a que estes os testem antes de se submeterem ao processo de certificação, diminuindo dessa forma a probabilidade de falhas dos componentes.

Uma vez criadas categorias de componentes, será necessário validar a presente metodologia num cenário maior do que o apresentado nesta dissertação. Esse cenário deve incluir vários componentes de cada categoria e avaliar vários cenários de integração, permitindo dessa forma uma avaliação mais consistente desta metodologia.

Fora do âmbito do AAL4ALL e mais orientada a um contexto académico, outra ideia interessante para ser desenvolvida no futuro, seria a análise à viabilidade de introduzir mais formalismo na especificação de categorias de produtos, por forma não só a retirar a ambiguidade, mas também tendo em vista a geração automática de testes.

Referências

- [AAL11] AAL4ALL. Ambient Assisted Living For All, 2011. URL: <http://www.aal4all.org/>.
- [AAL13] AAL4ALL. Definição de Modelo de Negócio e Gestão da Actividade, November 2013.
- [All13a] Continua Health Alliance. Certification Proces, June 2013. URL: <http://www.continuaalliance.org/products/certification-process>.
- [All13b] Continua Health Alliance. Continua Health Alliance - Official Website, June 2013. URL: <http://www.continuaalliance.org/>.
- [CCSS07] Randy Carroll, Rick Cnossen, Mark Schnell e David Simons. Continua: An Interoperable Personal Healthcare Ecosystem. *IEEE Pervasive Computing*, 6(4):90–94, October 2007.
- [Com13] European Commission. CE marking, June 2013. URL: http://ec.europa.eu/enterprise/policies/single-market-goods/cemarking/index_en.htm.
- [Eur13] EuroRec. Thematic Network on Quality Labelling And Certification of EHR Systems, June 2013. URL: <http://www.eurorec.org/rd/index.cfm>.
- [FLBM13] Joao Pascoal Faria, Bruno Lima, Tiago Boldt e Angelo Martins. A testing and certification methodology for an ambient-assisted living ecosystem. *Proceedings of the 15th IEEE International Conference on e-Health Networking, Applications and Services (Healthcom2013)*, pages 549–553, October 2013.
- [Fou13] Python Software Foundation. Python Programming Language – Official Website, June 2013. URL: <http://www.python.org/>.
- [Fun09] U.N.P. Fund. Population ageing: A larger and older population, June 2009. URL: <http://www.unfpa.org/pds/ageing.html>.
- [Gro11] Object Management Group. UML Superstructure Specification, 2011. URL: <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>.
- [HIT13] HITCH. The Healthcare Interoperability Testing and Conformance Harmonisation, June 2013. URL: <http://www.hitch-project.eu/about>.
- [IEE08] IEEE. IEEE Standard for Software and System Test Documentation. *IEEE Std 829-2008*, 2008.

REFERÊNCIAS

- [Ins07] European Telecommunications Standards Institute. Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic approach to interoperability testing, March 2007.
- [Ins13] European Telecommunications Standards Institute. Interoperability, June 2013. URL: <http://www.etsi.org/index.php/standards/interoperability>.
- [ISO96] ISO/IEC. General requirements for bodies operating product certification systems. *ISO/IEC Guide 65:1996*, 1996.
- [ISO98] ISO/IEC. General criteria for the operation of various types of bodies performing inspection. *ISO/IEC 17020:1998*, 1998.
- [ISO03] ISO/IEC. Conformity assessment – General requirements for bodies operating certification of persons. *ISO/IEC 17024:2003*, 2003.
- [ISO04a] ISO/IEC. Conformity assessment – Fundamentals of product certification. *ISO/IEC Guide 67:2004*, 2004.
- [ISO04b] ISO/IEC. Conformity assessment – General requirements for accreditation bodies accrediting conformity assessment bodies. *ISO/IEC 17011:2004*, 2004.
- [ISO04c] ISO/IEC. Conformity assessment – Guidance on a third-party certification system for products. *ISO/IEC Guide 28:2004*, 2004.
- [ISO04d] ISO/IEC. Conformity assessment – Supplier’s declaration of conformity – Part 1: General requirements. *ISO/IEC 17050-1:2004*, 2004.
- [ISO04e] ISO/IEC. Conformity assessment – Vocabulary and general principles. *ISO/IEC 17000:2004*, 2004.
- [ISO04f] ISO/IEEE. Health informatics – Point-of-care medical device communication – Part 20101: Application profiles – Base standard. *ISO/IEEE 11073-20101:2004*, 2004.
- [ISO05a] ISO/IEC. Conformity assessment – Guidance on the use of an organization’s quality management system in product certification. *ISO/IEC Guide 53:2005*, 2005.
- [ISO05b] ISO/IEC. General requirements for the competence of testing and calibration laboratories. *ISO/IEC 17025:2005*, 2005.
- [ISO06] ISO/IEC. Conformity assessment – Requirements for bodies providing audit and certification of management systems. *ISO/IEC 17021:2006*, 2006.
- [ISO08] ISO. Quality management systems – Requirements. *ISO 9001:2008*, 2008.
- [ISO09] ISO/IEC. Conformity assessment. General requirements for third-party marks of conformity. *ISO/IEC 17030:2009*, 2009.
- [LRL⁺08] Eirini Lekka, Harald Reiter, Jean Luprano, Reinhard Hammerschmidt e Nikos Maglaveras. Personalized Health Certification Procedure, March 2008. URL: http://heartcycle.med.auth.gr/resources/deliverables/D19.7_HeartCycle_FINAL.pdf.
- [Mic11] Onken Michael. Healthcare Interoperability Testing and Conformance Harmonisation, March 2011.

Anexo A

Artigos Produzidos

Ao longo do presente projeto de dissertação, foi produzido um artigo intitulado *A Testing and Certification Methodology for an Ambient-Assisted Living Ecosystem*, aceite e apresentado na *IEEE HealthCom'2013*, uma conferência internacional na área das tecnologias para a saúde e AAL. Uma cópia desse mesmo artigo é apresentada de seguida.

A Testing and Certification Methodology for an Ambient-Assisted Living Ecosystem

João Pascoal Faria

INESC TEC and Faculdade
de Engenharia da
Universidade do Porto
Porto, Portugal
jpf@fe.up.pt

Bruno Lima

INESC TEC and Faculdade
de Engenharia da
Universidade do Porto
Porto, Portugal
bruno.lima@fe.up.pt

Tiago Boldt

INESC TEC and Faculdade
de Engenharia da
Universidade do Porto
Porto, Portugal
tbsousa@inescporto.pt

Angelo Martins

INESC TEC and Instituto
Superior de Engenharia do
Porto
Porto, Portugal
amm@isep.ipp.pt

Abstract— To cope with the needs raised by the demographic changes in our society, several Ambient-Assisted Living (AAL) technologies have emerged in recent years, but those ‘first offers’ are often monolithic, incompatible and thus expensive and potentially not sustainable. The AAL4ALL project aims at improving that situation through the development of an open ecosystem of interoperable products and services for AAL, tied together via an integration infrastructure. To that end, the project encompasses the specification of a set of reference models and requirements for interoperable products and services, against which candidate products and services can be tested and certified, and subsequently integrated as components of the ecosystem. This paper proposes a testing and certification methodology for such an ecosystem.

Keywords—testing, certification, ambient-assisted living

I. INTRODUCTION

Developed societies are currently facing severe demographic changes: the world is getting older at an unprecedented rate. In 2000, about 420 million people (7% of the world’s population), were aged 65 or older. By 2050, that number will be nearly 1.5 billion people (16% of the world’s population). This demographic trend will be also followed by an increase of people with physical limitations. New challenges will be raised to the traditional systems of health care. There is an urgent need to find solutions that allow extending the time people can live in their preferred environment by increasing their autonomy, self-confidence and mobility.

Although some technologies for Ambient-Assisted Living (AAL) are already available and often in use as an answer to the needs stated above, these ‘first offers’ for primary and secondary end-users are monolithic, incompatible and thus expensive and potentially not sustainable.

The AAL4ALL project [1] aims at answering to those problems through the development of an ecosystem of interoperable products and services for AAL, associated to a business model and validated through a large scale trial. One goal of this project is to ensure that any supplier of AAL products and services, whether they are physical devices or software, can enter the ecosystem easily and independently, whilst assuring their interoperability with the rest of the ecosystem.

To that end, the AAL4ALL project encompasses the specification of a set of reference models and requirements for products and services (in the scope of work packages 1 to 4 in Figure 1), against which candidate products and services can be certified and subsequently integrated as components of the ecosystem. The project also encompasses the definition of a testing and certification methodology for these new components (in the scope of work package 5 in Figure 1), which is the subject of this paper.

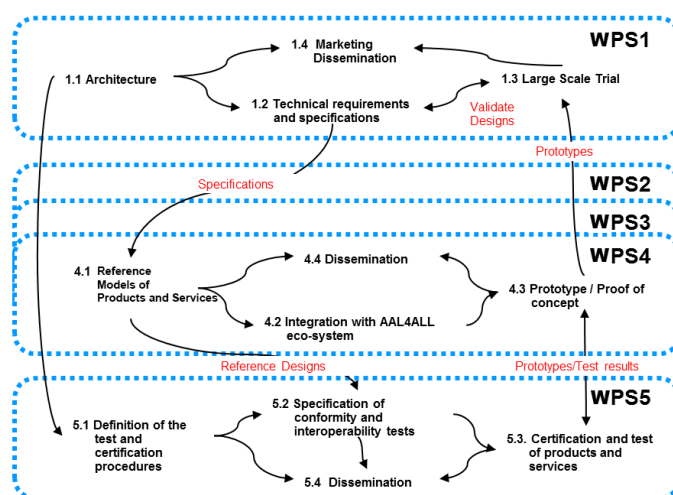


Figure 1 Structure of AAL4ALL project [1].

The main advantages of products and services certification and corresponding labelling are: assure their compliance with important technical regulations (for example for safety, compatibility, energy efficiency, environmental impact, conservation, and quarantine); assist regulatory bodies in their market surveillance of labeled products covered by their responsibility; assist manufacturers in the selection of certified components for their own products and facilitate the subsequent certification of their own assembled products; help suppliers and retailers selling their products; give consumers a higher confidence on the products they use and buy; enable interoperability in a multi-vendor, multi-network, multi-service environment, regarding different categories [2], such as technical, syntactical, semantic and organizational interoperability.

The main contribution of this paper is the definition of a novel testing and certification methodology, that was developed for the AAL4ALL ecosystem, but can be applied for other open socio-technical service systems.

The rest of the paper is organized as follows: section II presents the integration architecture for the AAL ecosystem envisioned in the AAL4ALL project; section III presents the proposed testing and certification lifecycle processes; section IV presents a meta-model with the main concepts involved; section V presents an example; related work is presented in section VI; conclusions and future work are drawn in section VII.

II. AAL ECOSYSTEM INTEGRATION ARCHITECTURE

Being a joint project by a consortium of 32 Portuguese partners from industry and academia and aiming at the development of an open AAL ecosystem by providing an infrastructure for third parties to integrate their sensors and services, AAL4ALL required a generic software platform to orchestrate how information flows in the ecosystem and to be able to manage and identify users and services in it.

Components to integrate with the system are one of sensors, actuators and services; these can be categorized according to their behavior as publishers (sensors and services) and subscribers (actuators and services) of information. Categorizing components this way allowed the orchestration to adopt the publisher-subscriber pattern to manage how information flows. Such orchestration is handled by clustered cloud-based components, responsible for forwarding published data to the proper subscribers, taking into account security, authorization, authentication and performance (through automatic scaling processes), which we call *AAL4ALL Nodes*.

AAL4ALL Operators are responsible for installing required components at the patient's house and connect them with the proper AAL4ALL Node, as well as providing either a subscription to a caretaking service that would provide remote monitoring by professionals or directly raise alarms to a contact person via email or SMS.

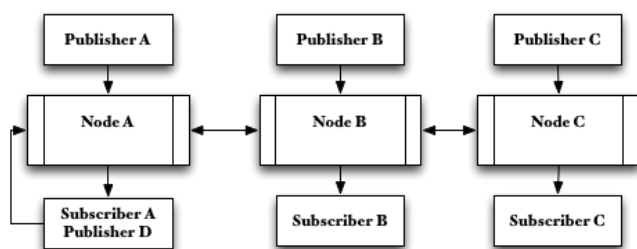


Figure 2 AAL4ALL nodes orchestrate information flows between third-party publisher and/or subscriber components.

Figure 2 shows three AAL4ALL Nodes clustered, with three publishers of data on top and three subscribers on the bottom, one of which is also publisher. DNS round robin distributes the load across nodes. In this scenario, Publisher A always sends its data to Node A. If Publisher A published data required by publisher C, Node A would propagate it to Node C for delivery.

Due to the open nature of this ecosystem, allowing any partner to provide his or her own components, certification is required to guarantee proper integration with the system.

III. STANDARDIZATION, TESTING AND CERTIFICATION PROCESS

Figure 3 presents the overall AAL4ALL standardization, testing and certification process.

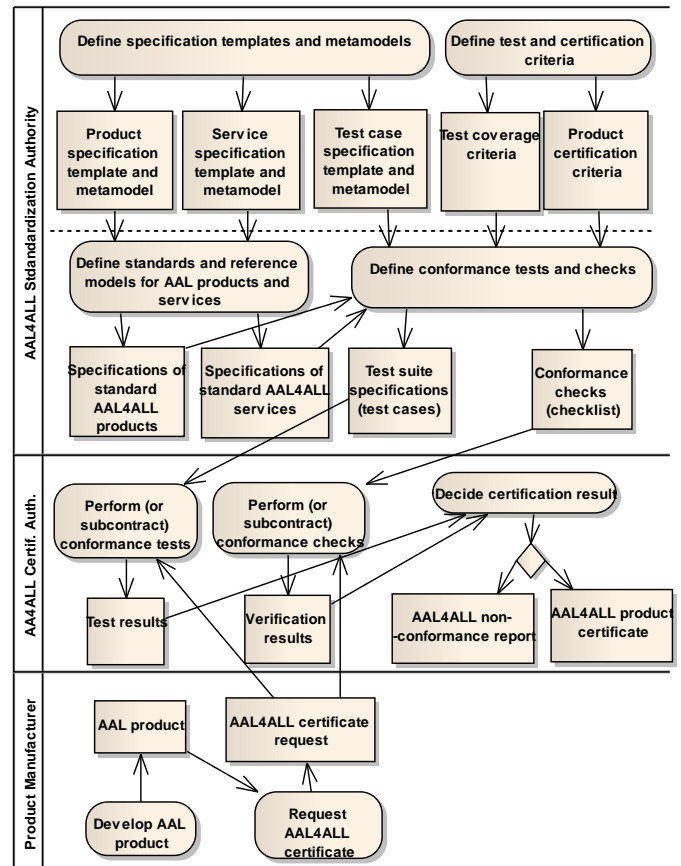


Figure 3 UML activity diagram describing the overall AAL4ALL standardization, testing and certification process.

Specifications of standard AAL products and services are created by an AAL4ALL Standardization Authority, currently represented by the project consortium (under work packages 2, 3 and 4). The specifications produced must obey common templates and metamodels, to be explained in section IV.

The testing and certification methodology encompasses the definition of common test and certification criteria that guide the derivation of conformance test suites and verification checklists for each standard AAL product or service, as will be explained in the following sections. These activities are also the responsibility of the AAL4ALL Standardization Authority.

The certification process starts with a request from a Product Manufacturer to an AAL4ALL Certification Authority, which is responsible to perform the defined tests and verification checks for the candidate product, and decide to issue a non-conformance report or a certificate (verifiable at runtime).

Currently, the Certification Authority is represented by the project consortium, under work package 5.

IV. METAMODELS AND CRITERIA

Figure 4 presents an overview of the metamodel proposed for the specification of standard AAL products and services (named *ComponentClass* and *ServiceClass*, respectively, in the figure) and test cases.

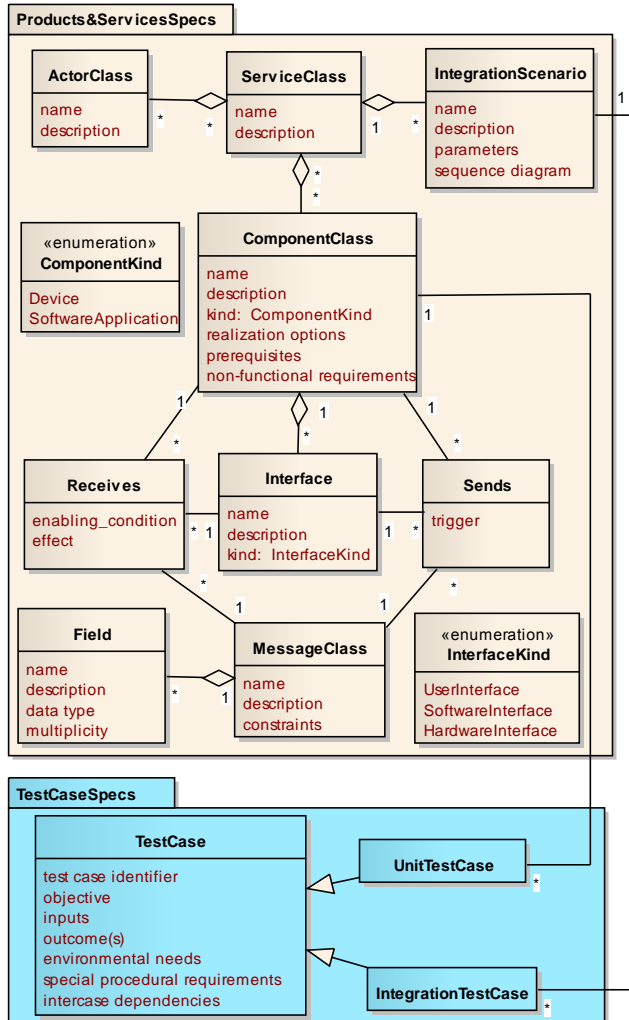


Figure 4 UML class diagram partially describing the metamodel for the specification of standard AAL products and services and associated test cases.

In this metamodel, a service should be understood as an end-to-end business service, involving a set of actors (like care receivers and providers) and components (devices or software applications), which interaction is described for key scenarios via UML sequence diagrams. This should not be confused with a software service.

Standard AAL components are described by the interfaces they expose (user interfaces or software interfaces), the messages they are able to send and receive through those interfaces, the events that trigger the sending of messages (e.g., receiving a message, reaching an internal state, or a time event), the effects of messages received (e.g., changing internal state,

or sending a message), the conditions upon which messages may be received (e.g., being in some internal state), and prerequisites they must satisfy (e.g., CE labeling, compliance with ISO/IEEE 11073 standard [7]). Messages are described by their name, fields and constraints on field values. Test cases for unit (component) and integration (scenario) testing are described according to the IEEE Std 829™-2008 [8].

Regarding the testing and certification criteria, the certification process of a candidate component involves three verification steps:

- 1) Checking the fulfillment of the standard component prerequisites via the analysis of appropriate evidences.
- 2) Passing units test cases derived from the standard component specification, covering all the messages, effects, conditions and triggers.
- 3) Passing integration test cases derived from the service integration scenarios in which the standard component participates (using certified components for the other participants), covering all the messages and branches.

V. EXAMPLE

To better understand how the concepts presented can be instantiated, we present a concrete example for an AAL monitoring service. As illustrated in Figure 5, the scenario consists of a Care Receiver who has two sensors (Balance and ECG) and a networking device (Home Gateway) at home. Somewhere in the cloud, an AAL4ALL NODE assures the communication between the Home Gateway and the web and mobile monitoring applications used by a Healthcare Provider and a Healthcare Informal Provider, respectively, according to a publish-subscribe model. The Healthcare Provider represents an entity (doctor, nurse, organization) that is responsible for monitoring and responding to any problems that may arise with the Care Receiver, while the Healthcare Informal Provider is a non-specialized person (family, friend) that wants to be informed about the status of the Care Receiver.

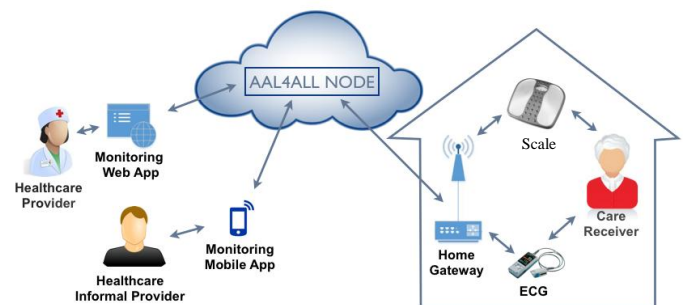


Figure 5 Example configuration for a AAL monitoring service

In this example, it is possible to identify different communication scenarios, one of which is illustrated in Figure 6. This scenario describes the mechanism for assuring that the Healthcare Provider receives an alert when the heart rhythm of the Care Receiver is outside a specified range. To simplify the description, it is assumed that the system has been previously configured and the devices are already able to communicate with each other.

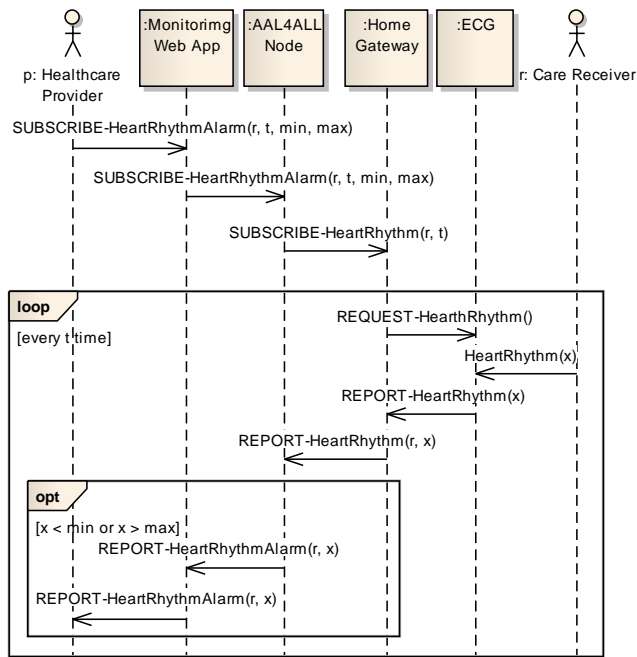


Figure 6 UML sequence diagram for an integration scenario S1.

Table 1 Partial example of a component specification

Name	Home Gateway
Description	Manages communication with in-home devices (sensors and actuators) and AAL4ALL Nodes.
Kind	Device
Interfaces	<div> name: I1 description: Interface with AAL4ALL Node. kind: SoftwareInterface </div> <div> name: I2 description: Interface with sensors and actuators. kind: HardwareInterface. </div>
State variables	Table of subscriptions
Receives	<div> message: SUBSCRIBE-HeartRhythm(r, t) interface: I1 enabl.cond.: always enabled effect: every t time do send REQUEST-HeartRhythm() to ECG of Care Receiver r </div> <div> message: REPORT-HeartRhythm(x) interface: I2 enabl. cond.: a REQUEST was sent effect: send REPORT-HeartRhythm(r, x), where r is the Care Receiver associated with the ECG sensor </div>
Sends	<div> message: REQUEST-HeartRhythm() interface: I2 trigger: see SUBSCRIBE-HeartRhythm(r, t) </div> <div> message: REPORT-HeartRhythm(r, x) interface: I1 trigger: see REPORT-HeartRhythm(x) </div>
Realization options	Communication. protocols at I2: Wi-Fi, Bluetooth, Ethernet
Prerequisites	CE marking
Non-funct. requirements	Message sending delay is less than 5 seconds.

Table 2 Partial example of unit test case specification.

Test case identifier	HG.1
Objective	Check if a home gateway, when requested to provide HeartRhythm information about a specified user r with a specified periodicity t , sends requests for HeartRhythm information to the appropriate user equipment with the appropriate timing.
Inputs	Configuration information mapping user identifiers and equipment identifiers. SUBSCRIBE-HeartRhythm(r, t)
Outcome(s)	REQUEST-HeartRhythm() messages, to the correct device, with t time period, with a maximum delay of 5 sec.
Environmental needs	Message injection and monitoring equipment.
Special procedural requirements	Before sending the SUBSCRIBE message, the gateway must be configured.
Intercase dependencies	-

Table 3 Partial example of integration test case specification.

Test case identifier	S1.1
Objective	Exercise a candidate component in scenario S1, and check the correct end-to-end behavior from the actors' perspective, covering all the scenario messages and conditions.
Inputs	<p>Form healthcare provider: Request heart rhythm alarms via the Monitoring Web App, regarding a given person r, with a certain periodicity t, for values outside specified min and max values.</p> <p>Form healthcare receiver r: provide heart rhythm information via attached ECG device.</p>
Outcome(s)	To healthcare provider: Alarm reports whenever appropriate.
Environmental needs	Devices, applications and actors needed to fulfill the scenario, plus a information recorder near the monitored person. Depending on the kind of ECG used, a separate device may be needed to show the user heart rhythm.
Special procedural requirements	<p>Healthcare receiver should alternate between rest periods with low heart rate and high activity periods with high heart rate (e.g., running).</p> <p>Experiment with different time periods (t), and heart rate ranges (min and max).</p> <p>Compare values (x) collected near the user and the Web App.</p> <p>All but the candidate component should be certified components (with the goal of testing the non-certified component for this scenario).</p>
Intercase dependencies	-

A sample component specification is shown in Table 1. The template used is based on the metamodel of Figure 4. Examples of test cases derived from the component specification and the integration scenario, according to the criteria described in section IV and the metamodel of Figure 4, are presented in Table 2 and Table 3.

VI. RELATED WORK

In order to try to solve the problems described at the beginning of this paper, the emergence of projects concerned with the standardization and interoperability in the eHealth area has been growing. Some of these projects arise not only through consortia of companies that produce products for the health sector, but also by institutions such as the European Union.

The Continua Health Alliance [3] is a non-profit, open industry alliance of healthcare and technology companies in the world joining together in collaboration to improve the quality of personal healthcare. Through the efforts of a collaborative industry organization, Continua aims at enabling a personal health ecosystem where many diverse vendors can combine their products into new value propositions with significant health benefits for people worldwide. One of the differences with the current project is that the Continua Health Alliance is focused on user devices and home communication, while the AAL4ALL project scopes end-to-end services, including also the communication with applications from the caretaker side, which interfaces are also subject to certification, and non-health related devices. The other difference is that the AAL4ALL project also includes the creation of an open integration infrastructure, enabling the interoperation of certified components, information sharing, service composition, an easy addition of new information providers and consumers. Of course, Continua standards may be used as prerequisites in our project whenever applicable.

EHR-Q^{TN} [4] is a Thematic Network project that prepares the health community across Europe for systematic and comparable quality assurance and certification of e-Health products, more specifically of Electronic Healthcare Record (EHR) systems. Although having a different focus, the EHR-Q^{TN} project produced a set of recommendations that may be applied also for the AAL ecosystem regarding the quality labelling and certification procedures, namely, the recommenda-

tions for third party assessments, start small, and incentivised model [5].

The Healthcare Interoperability Testing and Conformance Harmonisation (HITCH) project [6], developed in the 2010-2011 period, produced a set of recommendations to the EU Commission on how to proceed with eHealth interoperability testing and certification/labelling in Europe. A major recommendation is the development of a common European testing and certification foundation for interoperability in healthcare systems, upon which national or regional eHealth projects across Europe may organize their own testing and certification procedures in the future. Although such a common foundation could be very useful for our project, as far as we know, it was not yet created.

VII. CONCLUSIONS AND FUTURE WORK

It was presented an approach for the standardization, testing and certification of interoperable products and services for an AAL ecosystem that is being developed in the context of a nation-wide project. The approach is currently being applied and validated for real-world pilot scenarios.

As future work, we intend to set up a testing laboratory to partially automate integration and unit testing of candidate products, and apply the approach (from standardization to certification) to further AAL products.

REFERENCES

- [1] AAL4ALL – Official Website - <http://www.aal4all.org/>.
- [2] "Achieving Technical Interoperability – The ETSI Approach", Hans van der Veer(Alcatel Lucent), Anthony Wiles (ETSI Secretariat), 3rd edition – April 2008.
- [3] Continua Health Alliance – Official Website - <http://www.continuaalliance.org/index.html>.
- [4] Thematic Network on Quality Labelling And Certification of EHR Systems – Official Website - <http://www.eurorec.org/rd/index.cfm>
- [5] HITCH Key Messages. HITCH Team, November, 2011 - <http://www.hitch-project.eu/sites/www.hitch-project.eu/files/HITCH-Key-Messages.pdf>.
- [6] The Healthcare Interoperability Testing and Conformance Harmonisation project – Official Website - <http://www.hitch-project.eu/>.
- [7] ISO/IEEE 11073-20101:2004 Health informatics -- Point-of-care medical device communication -- Part 20101: Application profiles -- Base standard, ISO/IEEE, 2004
- [8] IEEE Std 829™-2008 - IEEE Standard for Software and System Test Documentation, IEEE Computer Society, 2008

Anexo B

Plataforma de automatização de testes sobre o componente AALNode

De forma automatizar o processo de execução de teste sobre o componente AALNode, foi elaborada uma plataforma de automatização baseada em *scripts* criados em *python*. Para além dos ficheiros principais, apresentados no capítulo 4 essa plataforma contempla os seguintes *scripts*:

```
1 def SubMenu():
2     os.system('clear')
3     loop = 1
4     choice = 0
5
6     print (30 * '-')
7     print ("    AALNode Test - M E N U")
8     print (30 * '-')
9     print ("1. Run Nod.T1")
10    print ("2. Run Nod.T2")
11    print ("3. Run Nod.T3")
12    print ("4. Run Nod.T4")
13    print ("5. Run Nod.T5")
14    print ("6. Run Nod.T6")
15    print ("7. Run Nod.T7")
16    print (30 * '-')
17
18    while loop == 1:
19        ## Get input ###
20        choice = raw_input('Enter your choice [1-7] : ')
21
22        ### Convert string to int type ##
23        choice = int(choice)
24        loop = 0
25
26    if choice == 1:
27        subprocess.call(["python", "NodT1.py"])
```

```

28     elif choice == 2:
29         subprocess.call(["python", "NodT2.py"])
30     elif choice == 3:
31         subprocess.call(["python", "NodT3.py"])
32     elif choice == 4:
33         subprocess.call(["python", "NodT4.py"])
34     elif choice == 5:
35         subprocess.call(["python", "NodT5.py"])
36     elif choice == 6:
37         subprocess.call(["python", "NodT6.py"])
38     elif choice == 7:
39         subprocess.call(["python", "NodT7.py"])
40     else:    ## default ##
41         print ("Invalid number. Try again...")
42         loop = 1
43
44
45 def Menu():
46
47     os.system('clear')
48
49     loopM = 1
50     choiceM = 0
51
52     print (30 * '-')
53     print ("    AALNode Test - M E N U")
54     print (30 * '-')
55     print ("1. Run All Tests")
56     print ("2. Run a particular test")
57     print (30 * '-')
58
59     while loopM == 1:
60         ## Get input ###
61         choiceM = raw_input('Enter your choice [1-2] : ')
62
63         ### Convert string to int type ##
64         choiceM = int(choiceM)
65         loopM = 0
66
67
68         ### Take action as per selected menu-option ###
69         if choiceM == 1:
70             print ("Starting run all tests...")
71             subprocess.call(["python", "NodT1.py"])
72             subprocess.call(["python", "NodT2.py"])
73             subprocess.call(["python", "NodT3.py"])
74             subprocess.call(["python", "NodT4.py"])
75             subprocess.call(["python", "NodT5.py"])
76             subprocess.call(["python", "NodT6.py"])

```



```
77     subprocess.call(["python", "NodT7.py"])
78     elif choiceM == 2:
79         SubMenu()
80     else:     ## default ##
81         print ("Invalid number. Try again...")
82         loopM = 1
83
84
85 Menu()
```

Listagem B.1: Script do Menu

```
1  TestName = 'NodT1'
2  print "Starting run " + TestName + "...
3
4  # Create Expected File
5  CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload"}']])
6
7  # Activate Subscribers
8  Subscriber(1, ['health.devices.1'], TestName, FDG)
9
10 # Send Menssages
11 Sender(1, 'health.devices.1', '{"data":"binary payload"}', FDG)
12
13 # wait 200 ms
14 time.sleep(0.2)
15
16 # Check Result
17
18 # print the test result
19 if (checkresult(1, TestName) == 0):
20     print TestName + " PASSED! All messages have been successfully received by
        their subscribers"
21 else:
22     print TestName + " FAILED! Try debug mode (change FDG to 1) for more
        informations"
```

Listagem B.2: Script do teste Nod.T1

```
1 TestName = 'NodT2'
2 print "Starting run " + TestName + "...
3
4 # Create Expected File
5 CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 1"}'], ['
    health.devices.1', '{"data":"binary payload 2"}'], ['health.devices.1', '{"data
    ":"binary payload 3"}']])
6
7 # Activate Subscribers
8 Subscriber(1, ['health.devices.1'], TestName, FDG)
9
10 # Send Menssages
11 Sender(1, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
12 Sender(2, 'health.devices.1', '{"data":"binary payload 2"}', FDG)
13 Sender(3, 'health.devices.1', '{"data":"binary payload 3"}', FDG)
14
15 # wait 200 ms
16 time.sleep(0.2)
17
18 # Check Result
19
20 # print the test result
21 if (checkresult(1, TestName) == 0):
22     print TestName + " PASSED! All messages have been successfully received by
        their subscribers"
23 else:
24     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in
        temp and ExpFiles folders for more informations"
```

Listagem B.3: Script do teste Nod.T2

```
1  TestName = 'NodT3'
2  print "Starting run " + TestName + "... "
3
4  # Create Expected File
5  CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 1"}']]
6  CreatExpFile (TestName, 2, [['health.devices.1', '{"data":"binary payload 1"}']]
7  CreatExpFile (TestName, 3, [['health.devices.1', '{"data":"binary payload 1"}']]
8
9  # Activate Subscribers
10 Subscriber(1, ['health.devices.1'], TestName, FDG)
11 Subscriber(2, ['health.devices.1'], TestName, FDG)
12 Subscriber(3, ['health.devices.1'], TestName, FDG)
13
14 # Send Menssages
15 Sender(1, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
16
17
18 # wait 200 ms
19 time.sleep(0.2)
20
21 # Check Result
22
23 # print the test result
24 if (checkresult(3, TestName) == 0):
25     print TestName + " PASSED! All messages have been successfully received by
        their subscribers"
26 else:
27     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in
        temp and ExpFiles folders for more informations"
```

Listagem B.4: Script do teste Nod.T3

```
1 TestName = 'NodT4'
2 print "Starting run " + TestName + "..."
3
4 # Create Expected File
5 CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 1"}']]
6 CreatExpFile (TestName, 2, [['health.devices.2', '{"data":"binary payload 1"}']]
7 CreatExpFile (TestName, 3, [['health.devices.3', '{"data":"binary payload 1"}']]
8
9 # Activate Subscribers
10 Subscriber(1, ['health.devices.1'], TestName, FDG)
11 Subscriber(2, ['health.devices.2'], TestName, FDG)
12 Subscriber(3, ['health.devices.3'], TestName, FDG)
13
14 # Send Menssages
15 Sender(1, 'health.devices.4', '{"data":"binary payload 1"}', FDG)
16 Sender(2, 'health.devices.2', '{"data":"binary payload 1"}', FDG)
17 Sender(3, 'health.devices.3', '{"data":"binary payload 1"}', FDG)
18 Sender(4, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
19
20
21 # wait 200 ms
22 time.sleep(0.2)
23
24 # Check Result
25
26 # print the test result
27 if (checkresult(3, TestName) == 0):
28     print TestName + " PASSED! The message was correctly rejected!"
29 else:
30     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in
        temp and ExpFiles folders for more informations"
```

Listagem B.5: Script do teste Nod.T4

```
1  TestName = 'NodT5'
2  print "Starting run " + TestName + "..."
3
4  # Create Expected File
5  CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 2"}']])
6
7  # Send Menssages
8  Sender(1, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
9
10 # Activate Subscribers
11 Subscriber(1, ['health.devices.1'], TestName, FDG)
12
13 # Send Menssages
14 Sender(1, 'health.devices.1', '{"data":"binary payload 2"}', FDG)
15
16 # wait 200 ms
17 time.sleep(0.2)
18
19 # Check Result
20
21 # print the test result
22 if (checkresult(1, TestName) == 0):
23     print TestName + " PASSED! All messages have been successfully received by
        their subscribers"
24 else:
25     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in
        temp and ExpFiles folders for more informations"
```

Listagem B.6: Script do teste Nod.T5

```
1 TestName = 'NodT6'
2 print "Starting run " + TestName + "..."
3
4 # Create Expected File
5 CreatExpFile (TestName, 1, [['health.devices.1', '{"data":"binary payload 1"}'], ['
    health.devices.1', '{"data":"binary payload 1"}']])
6
7 # Activate Subscribers
8 Subscriber(1, ['health.devices.1'], TestName, FDG)
9
10 # Send Menssages
11 Sender(1, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
12 Sender(2, 'health.devices.1', '{"data":"binary payload 1"}', FDG)
13
14 # wait 200 ms
15 time.sleep(0.2)
16
17 # Check Result
18
19 # print the test result
20 if (checkresult(1, TestName) == 0):
21     print TestName + " PASSED! All messages have been successfully received by
        their subscribers"
22 else:
23     print TestName + " FAILED! Try debug mode (change FDG to 1) or check files in
        temp and ExpFiles folders for more informations"
```

Listagem B.7: Script do teste Nod.T6

Anexo C

Descrição dos componentes que integram o cenário de teste

As informações dos componentes que integram o cenário de teste apresentado no presente projeto de dissertação, foram extraídas da documentação dos mesmos segundo o *template* do projeto AAL4ALL.

De seguida são apresentados o *Template* de descrição de componentes do projeto AAL4ALL e o documento resultante da descrição do componente “AALNode” utilizando o *Template*

AAL4ALL

PADRÃO DE CUIDADOS PRIMÁRIOS PARA SERVIÇOS DE AAL

Description of <product name>

Editors:	<name>
Delivery Nature:	Report
Confidentiality Level:	Private
Contractual Delivery Date:	<dd/mm/yy>
Real Delivery:	
Recommended for:	All AAL4ALL project members
Version:	<version number>
Number of Pages:	<pages number>
Key Words:	<key words>

Technical Template

Disclaimer

[This document contains material, which is the copyright of certain AAL4ALL consortium parties, and may not be reproduced or copied without permission. All AAL4ALL consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information. Neither the AAL4ALL consortium as a whole, nor a certain party of the AAL4ALL consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.]

Impressum

[Project Name] Ambient Assisted Living for All

[Project Acronym] AAL4ALL

[Number and Title of PPS] PPS5

[Document Title] Technical Template

[Editor: Name, Co-promoter] Juliana Teixeira, M. António Pereira, UM/CCG

[PPS Leader: Name, Co-promoter] Angelo Martins, INESC Porto

[Estimation of effort expended in this report]

Copyright notice

© 2011 to 2013 AAL4ALL Project

Executive Summary

[This report intends to present the format for the product specification in AAL4ALL context. This should be used by the product suppliers to specify their products].

DRAFT

Authors List

[illegible]

Contributions

Leader	Chapter	Section	Sub-Section	Contributors
<leader name>	<chapter number>	<section number>	<sub-section number>	<contributors name>

Revision History

Co-Promoter	Changed Section / Revision Type	Author	Date
<co-promoter name>	<section name>	<author name>	<dd/mm/yy>

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
AUTHORS LIST	4
GLOSSARY	10
1. THE CONTEXT OF USE	11
1.1. Introduction	11
1.2. Overall Product Characterization	11
1.3. Product purpose	11
1.4. Context of Use	11
1.4.1. Functional and Non-Functional Requirements	12
1.4.2. Description of Users.....	12
1.4.3. Goals and Tasks	13
1.4.4. Use Cases.....	13
1.5. Requirements of Technical Environment.....	14
1.6. Users Training.....	15
1.6.1. Installation Manual.....	15
1.6.2. Utilization Manual	15
2. PRODUCT INTEGRATION DESCRIPTION FOR AAL4ALL.....	16
2.1. Introduction	16
2.2. Interoperability Requirements.....	16
2.3. Interoperability Architecture	16
2.4. Product Interfaces.....	18
2.5. Interoperability Protocols and Profiles.....	19
2.5.1. Interoperability Profiles.....	19
2.5.2. Message Communications Protocol	19
3. REFERENCES	21

LIST OF FIGURES

<i>FIGURE 1: USE CASES DIAGRAM</i>	<i>14</i>
<i>FIGURE 2: AAL4ALL STANDARD ARCHITECTURE</i>	<i>18</i>
<i>FIGURE 3: DIAGRAM OF SERVICE DETAILS</i>	<i>19</i>
<i>FIGURE 4: SEQUENCE DIAGRAM</i>	<i>20</i>

LIST OF TABLES

TABLE 1: PRODUCT CHARACTERIZATION 11

TABLE 2: PRODUCT FUNCTIONAL REQUIREMENTS.....12

TABLE 3: PRODUCT NON-FUNCTIONAL REQUIREMENTS.....12

TABLE 4: PRODUCT STAKEHOLDERS13

TABLE 5: PRIMARY GROUPS OF PRODUCT USERS13

TABLE 6: PRODUCT USERS GROUPS A)13

TABLE 7: PRIMARY GROUPS OF PRODUCT USERS13

TABLE 8: PRODUCT GOALS AND TASKS.....13

TABLE 9: USE CASE DESCRIPTION14

TABLE 10: TECHNICAL ASSUMPTIONS14

TABLE 11: DESIGN STAGES OF INTEROPERABILITY REQUIREMENTS16

TABLE 12: INFORMATION TABLE FOR MACRO ARCHITECTURE COMPONENTS.....16

Glossary

Purpose

<specify the purpose of this Glossary>

Scope

<brief description of the scope>

Definitions

[The terms defined here forms the essential substance of the document. They can be defined in any order desired, but generally alphabetical order provides the greatest accessibility.]

<aTerm>

<the definition for aTerm is presented here. As much information as the reader needs to understand the concept should be presented>

1. The Context of Use

1.1. Introduction

[Describe the main goals of the product use in the AAL4ALL context, the functional and non-functional requirements, describes users profiles and use cases corresponding. Also describes the requirements of technical environment and the installation and utilization manual.]

1.2. Overall Product Characterization

Table 1: Product Characterization

Name and Version	<i><Product name> , release <version number></i>
Main goal of the product	<i>[brief description of the main goal of the product]</i>
Intended user groups	<i>[brief list of intended user groups]</i>
Training required	<i>[identify if it is necessary the training of users]</i>
Service supported by product	<i>[describe the services that can be supported by product]</i>
Usage environment	<i>[describe the product usage environment]</i>
Users with special needs	<i>[list of special needs of users]</i>

1.3. Product purpose

[Describe the overall product purpose and a brief description of the product use in context AAL4ALL]

*< brief description of product purpose in the AAL4ALL context and specify the service provided by product>
<informal drawing>*

1.4. Context of Use

[A brief description of examples of scenarios of use, the intended users, their goals, associated equipment (including hardware, software, and materials).]

< description a scenario of utilization in the context AAL4ALL>

Product : *<product name>*

Service: *<service name>*

User	<i><user name> <describe de state health of user, concerns, and main physical limitations></i>
Goals	<i><describe the main goals of the user to overcome their limitations ></i>
Associated equipment (hardware, software and materials)	Hardware: <i><brief list of hardware></i> Software: <i><brief list of software></i>
Physical and social environment (ex.: humidity, temperature, etc)	<i><brief description of physical and social environment characteristics></i>

Table 2: Brief description of main items of the scenario of utilization

1.4.1. Functional and Non-Functional Requirements

[Describe functional and non-functional system requirements]

Table 3: Product Functional Requirements

Functional Requirements	
Requirements name	Description
<i><requirement name></i>	<i><brief description of requirement name></i>

Table 4: Product Non-Functional Requirements

Non-Functional Requirements	
Requirements name	Description
<i><requirement name></i>	<i><brief description of requirement name></i>

1.4.2. Description of Users

[Describe relevant characteristics of the users.]

a) Stakeholders

[Describe the groups of people who are identified as stakeholders.]

Table 5: Product Stakeholders

Stakeholders	Description	Roles and interests
<Stakeholder name>	<description about stakeholder>	< roles and interests in the product >

b) Users groups

[Describe the groups of people, who are identified as user groups, including primary group, secondary group and their description.

- *Primary group: Primary group play an important role in product utility and interacts directly with the product. The product existence depends of primary group.*
- *Secondary group: Secondary group interacts with product but the existence of the product does not depend of secondary group.]*

Table 6: Primary Groups of Product Users

Primary group	Description
<primary group name>	< description about primary group>

Table 8: Primary Groups of Product Users

Secondary group	Description
<secondary group name>	< description about secondary group>

1.4.3. Goals and Tasks

[List the main goals and tasks for each user group.]

Table 9: Product Goals and Tasks

User group	Goals	Tasks
<user group name>	<main goals>	<main tasks>

1.4.4. Use Cases

[Describe the main use cases of the product that should be represented in use cases diagrams such as represented in Figure 1. Use case detail description shall be following the structure of

Table 10.]

<UML use cases diagram>

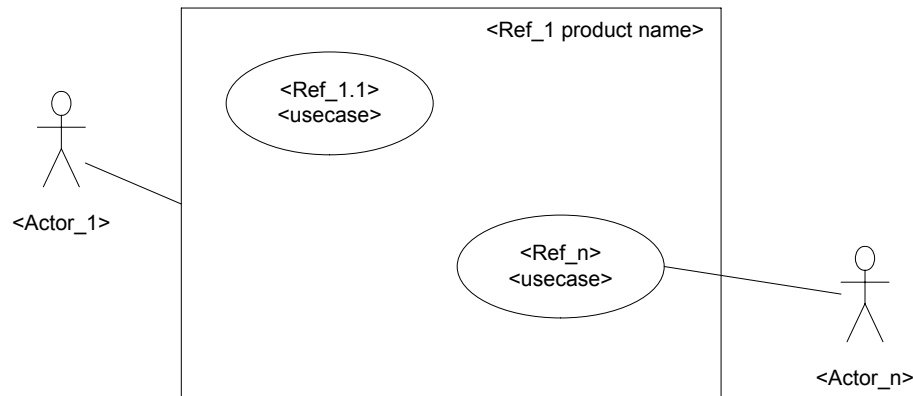


Figure 1: Use cases diagram

Table 10: Use Case description

Reference	<i><use case reference></i>
Name	<i><use case name></i>
Short Description	<i><use case description></i>
Actors	<i><list of users></i>
Basic Flow	<i><describe the basic flow of the use of product functionalities by the users></i>

1.5. Requirements of Technical Environment

[Describe the relevant requirements of the technical assumptions needed to the product execution, i.e., CPU, memory, S.O., Browsers, internet connection, LAN and WAN requirements]

Table 11: Technical Assumptions

Technical Assumptions	Description
Software requirements	<i><technical software requirements description></i>
Hardware requirements	<i><technical hardware requirements description></i>
Network requirements	<i><technical network requirements description></i>
Non-functional requirements	<i><non-functional requirements description></i>

1.6. Users Training

[Users are expected to study any documentation, training materials or courses before using the product. Normally, the most used training documents are the installation manual for installers and the utilization manual for users.]

1.6.1. Installation Manual

< describe the instructions for installation and configuration >

1.6.2. Utilization Manual

<describe the instructions for product utilization>

2. Product Integration Description for AAL4ALL

2.1. Introduction

[This section shall expose the description of the product integration, specifically aspects that will allow:

- *develop clear requirements;*
- *develop a comprehensive architectural overview, including clear identification of interoperable interfaces;*
- *Concentrate on specifying the right things, i.e. interoperable interfaces, and resist detailing internal implementation;*
- *use good protocol design techniques, such as*
 - o separation and description of normal behaviour and behaviour under error conditions;*
 - o full specification of options, including consequences of not implementing options;*
 - o development of (interoperability) profiles, where appropriate;*
 - o full specification of data (messages) and the encoding of that data;]*

2.2. Interoperability Requirements

[The typical objectives of an interoperability requirement are to: ensure the application or component interoperates with other specified applications and components: can pass necessary data to the other applications and components, receive necessary data from the other applications and components, can use the data it receives, can request the public services of the other applications and components and minimize integration defects.]

Table 12: Definition of interoperability requirements




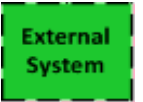
Interoperability requirements	<i>< specify the interoperability requirements ></i>
-------------------------------	------------------------------------------------------------

2.3. Interoperability Architecture

[Develop a comprehensive architectural overview, including clear identification of interoperable interfaces in AAL4ALL context, identifying the local where product can be applied taking in account the four macro architecture elements. The following table will present the information of the four macro architecture elements to take in count when develop the product architecture.]

Table 13: Information table for macro architecture component

[The following figure is the example of the standard architecture that shall be followed for the design of the product

Components	AAL Local Systems	AAL Gateway	AAL NODE	External Systems
Purpose	<i>[Product that is used in a user local environment]</i>	<i>[Product that is used as a Home or Mobile Gateway, or any component that can be integrated in the Gateways as a gateway functionality]</i>	<i>[Product that is used as a AAL NODE component or that can be integrated in the AAL NODE]</i>	<i>[Product that is used as an external system integrated in AAL context]</i>
Macro Architectural Elements Design				
Interfaces	<i>[Product interoperability interfaces at technical and syntactical levels between: - Local system and AAL Gateway - other systems if applicable]</i>	<i>[Product interoperability interfaces at technical and syntactical levels between: - AAL Gateway and Local system - AAL Gateway and AAL NODE - other systems if applicable]</i>	<i>[Product interoperability interfaces at technical and syntactical levels between: -AAL NODE and AAL Gateway - AAL NODE and External Systems other systems if applicable]</i>	<i>[Product interoperability interfaces at technical and syntactical levels between: -External Systems and AAL NODE -Other systems if applicable]</i>

architecture specified.

Remote area: Area composed for remote systems that are used to provide services to AAL ecosystem.

- *AAL Node - AAL4ALL Node is responsible for receiving, validating and delivering data in form of individual messages from where these are produced (producers) to those interested in processing them (consumers).*
- *External systems – Systems that are directly linked to the AAL Node that provides AAL services.*

Local area: Area composed for local systems that are used in local user environment (in-home or out-home).

- *AAL gateway- AAL Gateway (Home Gateway or Mobile Gateway) provides a set of functionalities that are executed locally to support the integration and execution of local systems.*
- *Local Systems - Software or System installed in a local user environment (in-home or out-home).]*

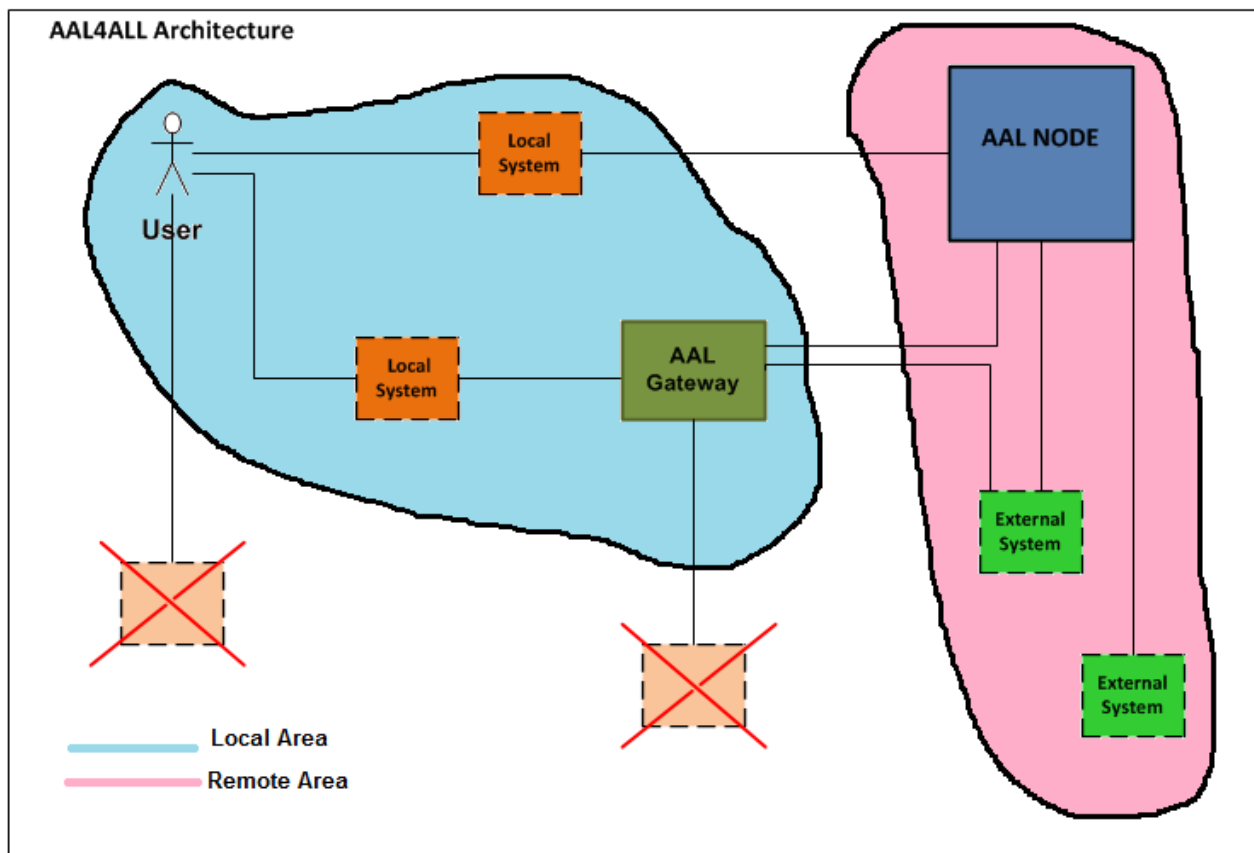


Figure 2: AAL4ALL standard Architecture

*< Identify the product according to the component of the architecture where the product is placed, i.e. **Local System** - corresponds to **Product A** (where Product A is the product described in chapter 1) >*

Table 14: Architecture components

Local System (If applicable)	
<i><product A></i>	<i><product description and interactions with other components></i>
External System (If applicable)	
<i><product B></i>	<i><product description and interactions with other components></i>

2.4. Product Interfaces

[The definition of interface standards aims at enabling product interoperability in AAL4ALL context. This is important to identify interoperability standards used in product development.]

<describe the product interfaces (user, software, hardware and communications level, when applicable) implemented by the product and describe the main interoperability standards used.>

Table 15: Interface specification

Interfaces
Name: <i><product interface name></i>
Description: <i><describe the product interface and their components></i>
Interface Kind: <i><ex.: User interface, software interface, hardware interface and communications level interface, when applicable ></i>
Interoperability standards: <i><refer the interoperability standards used></i>

2.5. Interoperability Protocols and Profiles

[Use good protocol design techniques. Describe communications interfaces; define message communications protocol and development of interoperability profiles.]

2.5.1. Interoperability Profiles

[Development of (interoperability) profiles. Interoperability profiles shall be described with the use of service oriented Architecture (SOA) UML profile (SOAML), like specified in the follow figure.]

<design the interoperability profile in SOAML>

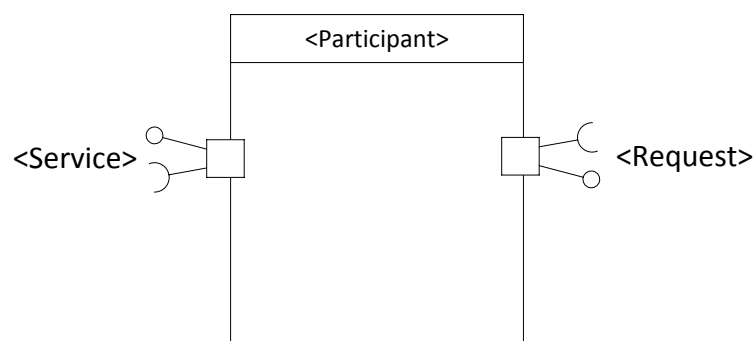


Figure 3: Diagram of service details

2.5.2. Message Communications Protocol

[Full specification of exchange of data (messages). The representation of protocol messages shall be done by the use of sequence diagrams, according to the follow example (Figure 4), and the table of message specification]

<design the sequence diagrams for message exchange between product integration AAL4ALL. >

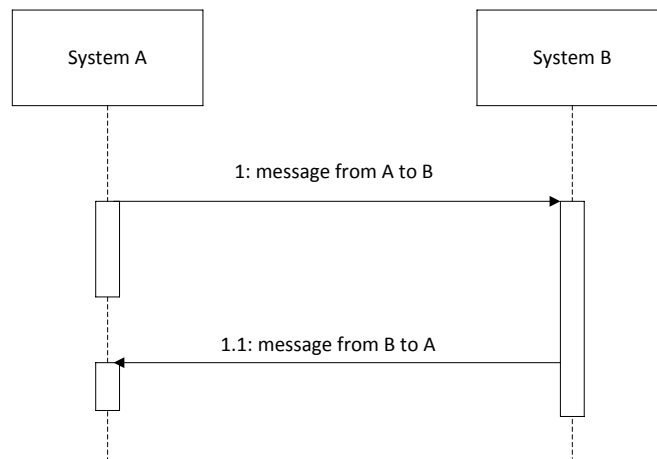


Figure 4: Sequence Diagram

Table 16: Message specification

Name	<i><describe the name of message></i>
General description	<i><describe the components where the message is sent and received></i>
Data Type Received	Message: <i><specify the message received></i> Interface: <i><mention the interface defined in section 2.4></i> Enable conditions: <i><specify the enable conditions></i> Effect: <i><specify the effect of message received></i>
Data Type Sent	Message: <i><specify the message sent></i> Interface: <i><mention the interface defined in section 2.4></i> Trigger: <i><specify the trigger></i>
Multiplicity	<i><refer the multiplicity of messages if applicable></i>

3. References

AAL4ALL

PADRÃO DE CUIDADOS PRIMÁRIOS PARA SERVIÇOS DE AAL

Description of AAL Node

Editors:	Tiago Boldt Sousa
Delivery Nature:	Report
Confidentiality Level:	Private
Contractual Delivery Date:	10/Oct/2013
Real Delivery:	
Recommended for:	All AAL4ALL project members
Version:	0.2
Number of Pages:	<pages number>
Key Words:	Node, Documentation, Certification

Technical Template

Disclaimer

[This document contains material, which is the copyright of certain AAL4ALL consortium parties, and may not be reproduced or copied without permission. All AAL4ALL consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information. Neither the AAL4ALL consortium as a whole, nor a certain party of the AAL4ALL consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.]

Impressum

[Project Name] Ambient Assisted Living for All

[Project Acronym] AAL4ALL

[Number and Title of PPS] PPS5

[Document Title] Technical Template

[Editor: Name, Co-promoter] Juliana Teixeira, M. António Pereira, UM/CCG

[PPS Leader: Name, Co-promoter] Angelo Martins, INESC Porto

[Estimation of effort expended in this report]

Copyright notice

© 2011 to 2013 AAL4ALL Project

Executive Summary

[This report intends to present the format for the product specification in AAL4ALL context. This should be used by the product suppliers to specify their products].

DRAFT

Authors List

[illegible]

Contributions

Leader	Chapter	Section	Sub-Section	Contributors
<leader name>	<chapter number>	<section number>	<sub-section number>	<contributors name>

Revision History

Co-Promoter	Changed Section / Revision Type	Author	Date
<co-promoter name>	<section name>	<author name>	<dd/mm/yy>

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
AUTHORS LIST	4
GLOSSARY	10
1. THE CONTEXT OF USE	11
1.1. Introduction	11
1.2. Overall Product Characterization	11
1.3. Product purpose	11
1.4. Context of Use	12
1.4.1. Functional and Non-Functional Requirements	12
1.4.2. Description of Users.....	13
1.4.3. Goals and Tasks	13
1.4.4. Use Cases.....	14
1.5. Requirements of Technical Environment.....	14
1.6. Users Training.....	15
1.6.1. Installation Manual.....	15
1.6.2. Utilization Manual	15
2. PRODUCT INTEGRATION DESCRIPTION FOR AAL4ALL.....	16
2.1. Introduction	16
2.2. Interoperability Requirements.....	16
2.3. Interoperability Architecture	16
2.4. Product Interfaces.....	17
2.5. Interoperability Protocols and Profiles.....	18
2.5.1. Interoperability Profiles.....	18
2.5.2. Message Communications Protocol	19
3. REFERENCES	23

LIST OF FIGURES

FIGURE 1: USE CASES DIAGRAM	14
FIGURE 2: AAL4ALL STANDARD ARCHITECTURE	17
FIGURE 3: DIAGRAM OF SERVICE DETAILS	19
FIGURE 4: SEQUENCE DIAGRAM	21

LIST OF TABLES

TABLE 1: PRODUCT CHARACTERIZATION 11

TABLE 2: PRODUCT FUNCTIONAL REQUIREMENTS.....12

TABLE 3: PRODUCT NON-FUNCTIONAL REQUIREMENTS..... 12

TABLE 4: PRODUCT STAKEHOLDERS 13

TABLE 5: PRIMARY GROUPS OF PRODUCT USERS **ERROR! BOOKMARK NOT DEFINED.**

TABLE 6: PRODUCT USERS GROUPS A) **ERROR! BOOKMARK NOT DEFINED.**

TABLE 7: PRIMARY GROUPS OF PRODUCT USERS **ERROR! BOOKMARK NOT DEFINED.**

TABLE 8: PRODUCT GOALS AND TASKS..... 13

TABLE 9: USE CASE DESCRIPTION 14

TABLE 10: TECHNICAL ASSUMPTIONS 15

TABLE 11: DESIGN STAGES OF INTEROPERABILITY REQUIREMENTS 16

TABLE 12: INFORMATION TABLE FOR MACRO ARCHITECTURE COMPONENTS**ERROR! BOOKMARK NOT DEFINED.**

Glossary

Purpose

<specify the purpose of this Glossary>

Scope

<brief description of the scope>

Definitions

[The terms defined here forms the essential substance of the document. They can be defined in any order desired, but generally alphabetical order provides the greatest accessibility.]

<aTerm>

<the definition for aTerm is presented here. As much information as the reader needs to understand the concept should be presented>

1. The Context of Use

1.1. Introduction

The AALNode is paramount in the AAL4ALL architecture, being responsible for all component integration in the ecosystem, where a component is any piece of software, either from a sensor, actuator, gateway or service. Working as a message bus, it adopts the publisher-subscriber design pattern, having data being published on it. Internal computation validates all received data in respect to its syntax, guaranteeing that wrongly formed messages are discarded. Valid messages are after forwarded to their proper destination(s).

1.2. Overall Product Characterization

Table 1: Product Characterization

Name and Version	AALNode , release 300913
Main goal of the product	Provide system integration capabilities, acquiring data from components and forwarding it to the proper destinations, according to internal rules and validations.
Intended user groups	The system is to be adopted by all component developers in the AAL4ALL ecosystem.
Training required	Developers should learn and comply with the available API for interacting with the system.
Service supported by product	System integration, through message publishing and subscribing.
Usage environment	The system should be deployed in the cloud, either as a standalone deployment, or behind a DNS load balancing system for clustered deployments.
Users with special needs	

1.3. Product purpose

AAL4ALL, as a possibly ultra-large scale system, might grow to reach hundreds of thousands of components that will be expected to exchange millions of messages per day. Furthermore, being an open ecosystem, constant change is expected, with clients having the ability to choose which services to subscribe to at a given time, easily switching between those available. The AALNode will be configured using the service subscription information, acquired from a remote service. This information will be paramount at configuring the routes for information delivered to a node.

1.4. Context of Use

Adoption of the AALNode will be transparent for final clients. Developers on the other hand will be requested to integrate all their components with it, being it sensors, actuators, gateways or remote services. By having all exchanged messages being routed through this system, logging, billing, monitoring and other management services can be developed independently and easily deployed in the ecosystem. Goals achieved through this approach include:

- **System Integration:** Any two components will be able to exchange messages, as long as the proper permissions have been granted between both parties.
- **Continuous system evolution:** Adding a new component to the ecosystem consists only of providing it with the proper authorization. Developers can independently evolve their own systems and deploy a new version without compromising the current version, gradually introducing changes.
- **Security:** All exchanged messages can be encrypted between the origin and final destination. Being message agnostic, the AALNode is responsible only for propagating the message to the proper destination.
- **Logging, Billing and Monitoring:** By imposing a central system for routing all messages in the system, external services can easily be integrated in the ecosystem to monitor all the traffic in it and evaluate it for multiple purposes, such as billing or statistical analysis.

1.4.1. Functional and Non-Functional Requirements

Table 2: Product Functional Requirements

Functional Requirements	
Requirements name	Description
Data Publication	Components should be able to publish data to the ecosystem.
Data Consumption	Data should be subscribed by components, following an ACL table.
Internal Validation	Badly formatted messages should be discarded right after reception.
Auto-configuration System	The AALNode should be able to use itself to acquire data from a remote service responsible for managing users subscriptions to services and keeping track of their devices. This information is paramount to impose proper message routing, allowing information to reach only authorized destinations.

Table 3: Product Non-Functional Requirements

Non-Functional Requirements	
Requirements name	Description

Performance	Messages should be routed in less than 200ms.
Security	For preventing access to restricted data, the system and server must be evaluated for possible security issues.
Availability	Downtime will result in ecosystem downtime and it must be prevented.
Fault tolerance	The system must have the ability to detect and recover from possible problematic scenarios
Safety	No unnecessary service should be running in the server, as it might present a security hazard.
Scalability	AALNodes are able to cluster for increased performance. It should be easy to start a new machine and have it join the processing cluster.
Testability	The system should be testable using black-box testing techniques to evaluate its behavior and performance as a whole.

1.4.2. Description of Users

a) Stakeholders

Table 4: Product Stakeholders

Stakeholders	Description	Roles and interests
Developers	Implements a product for the AAL4ALL ecosystem.	Integrates his system in the ecosystem by either having it publishing relevant information in the AALNode, or subscribing data from it.
System Administrator	Manages the platform where the system is running.	Monitor system load and scalability requirements. Deploy clustered instances for increased performance. Detect and report failures in the systems.

1.4.3. Goals and Tasks

[List the main goals and tasks for each user group.]

Table 5: Product Goals and Tasks

User group	Goals	Tasks
<i><user group name></i>	<i><main goals></i>	<i><main tasks></i>

1.4.4. Use Cases

[Describe the main use cases of the product that should be represented in use cases diagrams such as represented in Figure 1. Use case detail description shall be following the structure of

Table 6.]

<UML use cases diagram>

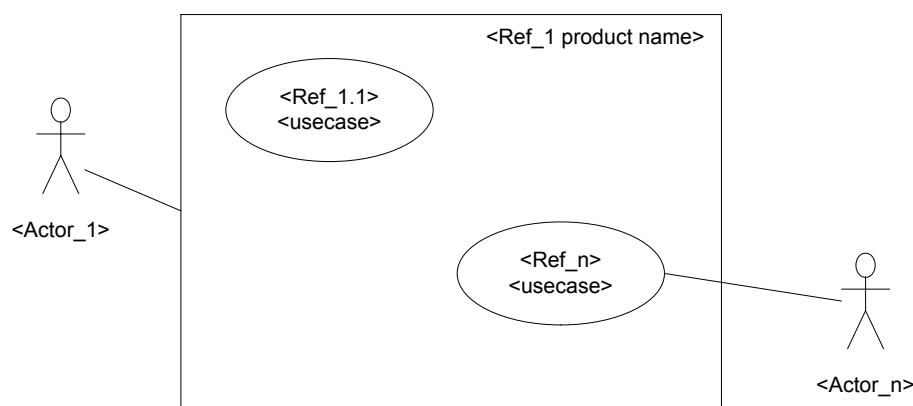


Figure 1: Use cases diagram

Table 6: Use Case description

Reference	<use case reference>
Name	<use case name>
Short Description	<use case description>
Actors	<list of users>
Basic Flow	<describe the basic flow of the use of product functionalities by the users>

1.5. Requirements of Technical Environment

AALNode is built for scalability. While a single deployment will be capable of handling the initial system load, future increase on the number of components in the ecosystem is expected. New nodes can be deployed in the same LAN as the original one and cluster with it, increasing the load handling capabilities. In such scenario, load-balancing systems would distribute requests amongst nodes in the cluster. Publishing can be performed to node, with messages being internally routed to all subscribers subscribed to that message. Requirements in the table bellow are relative to a single

network node.

Table 7: Technical Assumptions

Technical Assumptions	Description
Software	To install a single instance of the AALNode in a single machine, a computer with a recent distribution of Ubuntu Linux is required. Additionally, a JVM and an instance of RabbitMQ should also be available. All dependencies can be installed from the OS software repository.
Hardware	Requirements vary according to the expected load to handle in the machine. A minimal installation should provide a multi-core CPU and at least 4GB of RAM and 50GB of hard disk.
Network	This machine requires an internet connection with low latency. Latency is more relevant than speed. A symmetric broadband connection is recommended.

1.6. Users Training

Being a component for system integration, clients from the AAL4ALL project will never interact with the AALNode directly; no training is needed for end-users.

1.6.1. Installation Manual

The AALNode is composed by two systems. First, the data acquisition and validation module requires a Java Virtual Machine to be executed in. The second component consists of an installation of a RabbitMQ instance. Both modules can be replicated independently for scaling proposes. RabbitMQ instances should be deployed in the same Local Area Network, for clustering porpuses. Load balancing is achieved at in subscription using load balancing. A specialized team should perform deployment, being responsible for setting up the server machines and install all the required software to them.

1.6.2. Utilization Manual

The system is self-managed, meaning that no external intervention should be performed. In case of problematic behavior, INESC should be alerted. An administration team will gather the generated usage logs and trace the problem and fix the possible system defect.

2. Product Integration Description for AAL4ALL

2.1. Introduction

The AALNode is responsible for providing component integration in the AAL4ALL Ecosystem. It provides an API for components to publish their data using HTTP and a second one for subscribing to data using AMQP. This section describes its architecture and how components can integrate in the ecosystem by communicating with it.

2.2. Interoperability Requirements

Table 8: Definition of interoperability requirements

Interoperability requirements	<p>The node should have internet connection, which will use for both receiving and forwarding messages. All exchanged messages must follow the proper format or will otherwise be discarded.</p> <p>A remote service for managing users subscriptions and components must exist, from where the AALNode acquires its ACL configurations.</p>
-------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3. Interoperability Architecture

AAL Node - AAL4ALL Node is responsible for receiving, validating and delivering data in form of individual messages from where these are produced (publishers) to those interested in processing them (subscribers). It can connect with any component from the ecosystem. Figure 2 shows how all systems must connect via an AALNode to have their messages routed to the proper external systems.

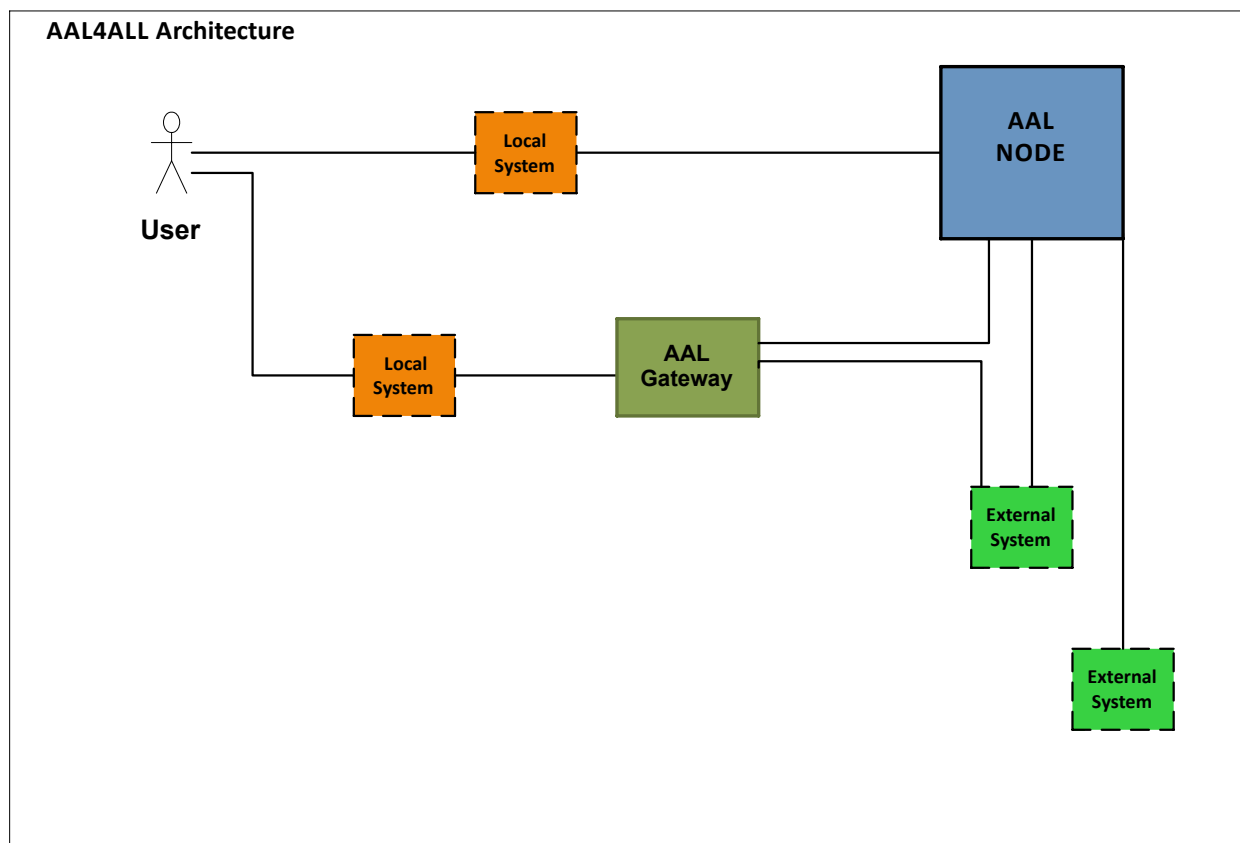


Figure 2: AAL4ALL standard Architecture

< Identify the product according to the component of the architecture where the product is placed, i.e. **Local System** - corresponds to **Product A** (where Product A is the product described in chapter 1) >

Table 9: Architecture components

Local System (If applicable)	
<product A>	<product description and interactions with other components>
External System (If applicable)	
<product B>	<product description and interactions with other components>

2.4. Product Interfaces

[The definition of interface standards aims at enabling product interoperability in AAL4ALL context. This is important to identify interoperability standards used in product development.]

<describe the product interfaces (user, software, hardware and communications level, when applicable) implemented by the product and describe the main interoperability standards used.>

Table 10: Interface specification

Interfaces
Name: Message Publish in an AALNode
Description: Messages are published to the AALNode using the HTTP protocol. All valid messages must be sent using the HTTP POST method with an authenticated request.
Interface Kind: Communication Level Interface
Interoperability standards: The HTTP protocol is the adopted standard for receiving messages in this interface.

Interfaces
Name: Message Subscription in an AALNode
Description: Messages are subscribed in the AALNode using the AMQP protocol. Subscription is performed using routing keys and user credentials (username/password), which validates the permission of the user when attempting to subscribe to new routing keys.
Interface Kind: Communication Level Interface
Interoperability standards: The AMQP protocol is the adopted standard for distributing messages in this interface.

2.5. Interoperability Protocols and Profiles

2.5.1. Interoperability Profiles

Interoperability in the AALNode is achieved using the two well-known HTTP and AMQP protocols, for both data publishing and subscription. Messages must be formatted using valid JSON, as described in section 2.5.2.

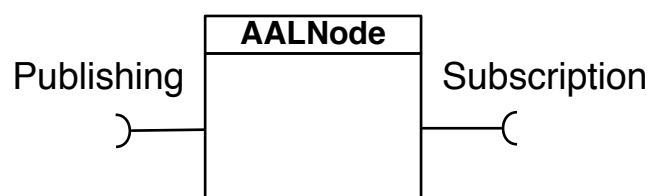


Figure 3: AALNode interoperability interfaces

2.5.2. Message Communications Protocol

The AALNode uses two different message formats: one that it expects to receive and another that it sends to subscribers, which is just a subset of the received message.

2.5.2.1. Published Messages

The input format, published by remote components to the AALNode, must be formatted using the JSON standard and include 4 fields:

origin: The unique key that identifies the sender and can be used to route traffic to it;

date: Unix time, corresponding to the number of seconds since 1 January 1970, of when this message was generated;

topic: Routing key, used to route the message to the proper subscriber(s);

payload: the content of the message, a JSON itself that might contain encrypted data for additional security.

The final message must be a valid JSON-formatted string. As such, some characters are invalid in the payload (such as single " or \). If a message is not valid JSON, it will be discarded on arrival. Otherwise, AALNode is agnostic to the payload content, meaning that publishers can create their own messages, as long as they respect the JSON format. Listing 1 exemplifies how a message can be build. Spaces can be omitted and are only used for better perception.

```
{
  "origin": "user.UserId",
  "date": "232425326235",
  "topic": "aalReminder.reminder.001",
  "payload": {
    "data": {
      "caretaker_name": "Dr.Bill",
      "patient_name": "John",
      "drug_name": "Brufen600",
      "dosage": "1pill",
      "message": "HelloJohn.Itistimetotakeyourmedicines.PleasetakeonepillofBrufen.",
      "video_url": {
        "video_url": "avatar.ccg.pt/videos/3ebff80cc5411e2b67d47fc50239af5.webm",
        "video_url_meobox":
"avatar.ccg.pt/videos/3ebff80cc5411e2b67d47fc50239af5.mp4"
      },
      "date": "2013060421: 02: 00"
    }
  }
}
```

Listing 1: A well-formatted message to be sent for the AALNode.

2.5.2.2. Routed Messages

Messages routed to subscribers are a subset of the received messages, containing only the content of the Payload field.

```
{
  "caretaker_name": "Dr.Bill",
  "patient_name": "John",
  "drug_name": "Brufen600",
  "dosage": "1pill",
  "message": "HelloJohn.Itistimetotakeyourmedicines.PleasetakeonepillofBrufen.",
  "video_url": {
    "video_url": "avatar.ccg.pt/videos/3ebcff80cc5411e2b67d47fc50239af5.webm",
    "video_url_meobox": "avatar.ccg.pt/videos/3ebcff80cc5411e2b67d47fc50239af5.mp4"
  },
  "date": "2013060421: 02: 00"
}
```

Listing 2: Message from listing 1 when forwarded to subscribers.

2.5.2.3. Message Orchestration

Orchestration, as shown in *Figure 4*, always occurs in a single direction: from publisher to subscriber. Whenever a message arrives at an AALNode, it is immediately forwarded to all subscribers who are currently subscribed to it. Messages with no subscribers or invalid are immediately discarded.

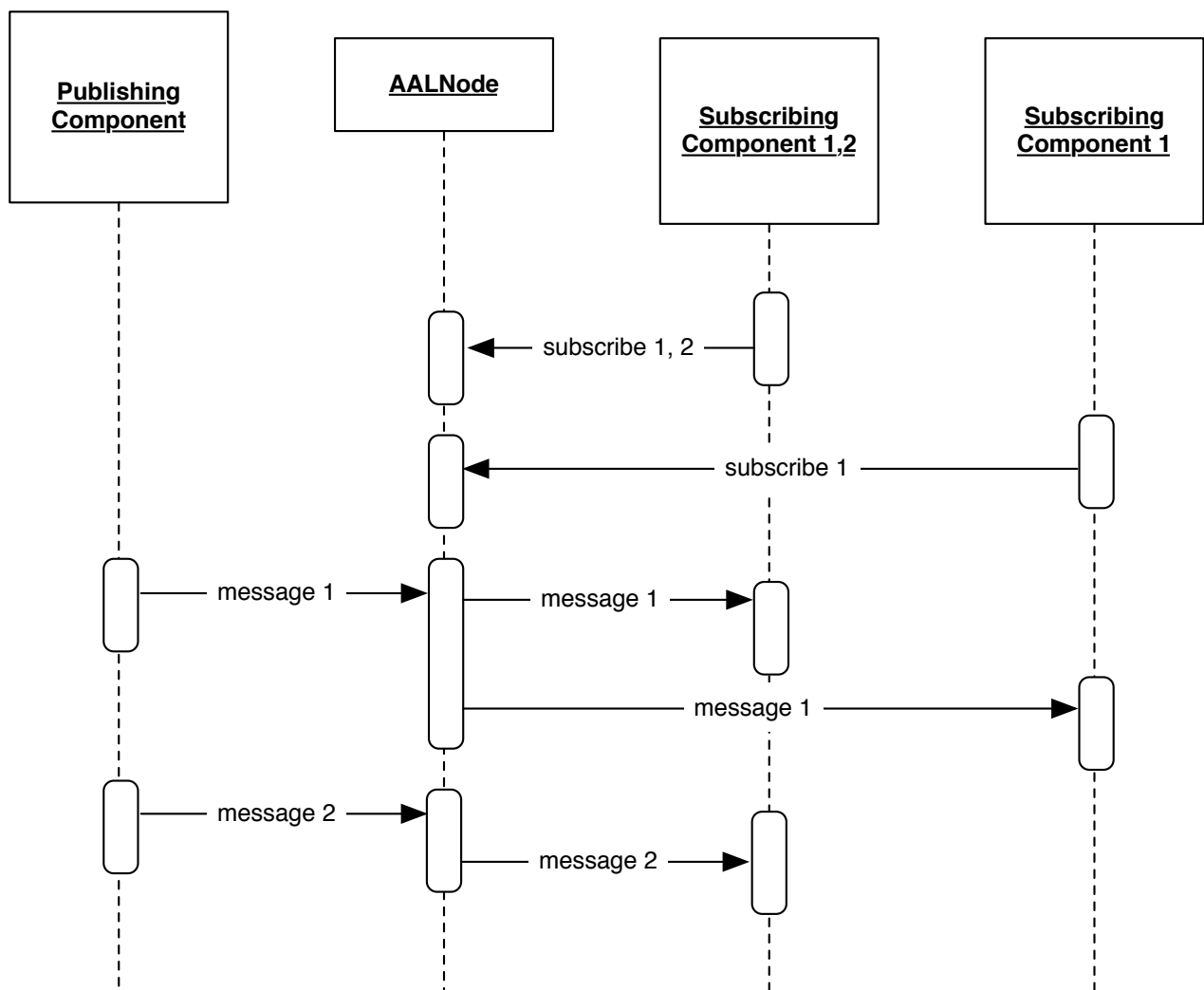


Figure 4: Message subscription, publishing and routing in the AALNode.

Table 11: Inbound Message specification

Name	Inbound AALNode Message.
General description	Message published by any component to the AALNode.
Data Type Received	Message: A valid JSON with previously described fields Interface: Message Publish in an AALNode
Data Type Sent	Message: "OK" or an error message/ Interface: <i>Message Publish in AALNode</i> Trigger: <i>Message successfully received</i>

Table 12: Outbound Message specification

Name	Outbound AALNode Message.
General description	Deliver message to subscribing components in the AALNode.
Data Type Sent	Message: The “payload” field of a received message. Interface: Message Subscription in AALNode Trigger: A valid message is received in the node and a component is subscribed to this message’s routing key.

3. References
